UIDE
Powered by
Arizona State University

Maestría en
SISTEMAS DE INFORMACIÓN

Mención **Inteligencia de Negocios y Analítica de Datos Masivos.**

Tesis previa a la obtención del título de Magíster en Sistemas de Información mención Inteligencia de Negocios y Analítica de Datos Masivos.

AUTOR: Ing. María Doménica Gómez Sánchez

Ing. Darío Hernán Herrera Herrera

Ing. Cristina Elizabeth Pavón Domínguez

Ing. Jacqueline Vanessa Quispe Tirado

TUTOR: Ing. Paulina Vizcaíno Ed.D

DISEÑO DE UN MODELO DE DETECCIÓN DE DATOS DUPLICADOS MEDIANTE PROCESAMIENTO DEL LENGUAJE NATURAL PARA OPTIMIZAR LA EFICIENCIA EN LA GESTIÓN DE DATOS EN UN LABORATORIO FARMACÉUTICO.

QUITO – ECUADOR | 2023

APROBACIÓN DEL TUTOR

Yo, Paulina Vizcaíno, certifico que conozco los autores/as del presente trabajo siendo los responsables exclusivos tanto de su originalidad y autenticidad, como de su contenido.

Ing. Paulina Vizcaíno Ed. D

DIRECTORA DE TESIS

CERTIFICACIÓN DE AUTORÍA

Nosotros, Ing. María Doménica Gómez Sánchez, Ing. Darío Hernán Herrera Herrera , Ing. Cristina Elizabeth Pavón Domínguez, Ing. Jacqueline Vanessa Quispe Tirado, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás disposiciones legales.

…...…….

Ing. María Doménica Gómez Sánchez

C.I.: 1718834227

…...……..……..…

Ing. Cristina Elizabeth Pavón Domínguez

C.I.: 1715287494

…...……

Ing. Darío Hernán Herrera Herrera

C.I.: 1718902529

…...……..……

Ing. Jacqueline Vanessa Quispe Tirado

C.I.: 1719895086

DEDICATORIAS

Con infinito amor y gratitud, a Dios y a la Virgen del Perpetuo Socorro, a mis Padres, a mis Viejitas, a mi Hermano. Todos y cada uno de mis logros pasados, presentes y futuros son por y para ustedes.

Ing. María Doménica Gómez Sánchez

Dedico este trabajo a Dios, a Jesús del Gran Poder, al Divino Niño, La Santísima Virgen María y al Hermano Miguel, por todas sus bendiciones que permitieron hacer realidad este anhelado sueño.

A mis padres Hernán y Blanca, que desde el cielo me siguen brindando todo su amor y cariño para seguir adelante.

A mi familia, por estar ahí siempre que los necesite apoyándome en todo sentido.

Ing. Darío Hernán Herrera Herrera

Dedico este trabajo de titulación a Dios, cuya fortaleza me ha sostenido a lo largo de este proceso de aprendizaje. En cada paso, siento su mano guiándome e iluminando mi camino.

A mi amado esposo Jefferson, a quien dedico este logro con todo mi amor. Tu constante respaldo ha sido mi mayor motivación. Este éxito también es tuyo, pues lo hemos construido juntos.

A mi querida madre Mery y a mi hermano Ronny, dedico este logro como expresión de gratitud por su apoyo incondicional, su amor infinito y por ser la razón detrás de cada logro.

Ing. Jacqueline Vanessa Quispe Tirado

Dedico este trabajo a mi esposo, cuyo constante estímulo me enseño que la pasión y la dedicación pueden trascender las barreras de cualquier campo profesional. Gracias por ser quien enciende mi búsqueda constante de conocimiento en diferentes áreas.

A mi madre Rita y mis hermanas Fernanda y Joselin, quienes me enseñan día a día que, a pesar de los logros profesionales y retos, la familia es primero.

A mis socios de vida y emprendimientos Mary y Tonny, quienes han confiado y respaldado mis ideas, gracias a sus conocimientos hemos logrados conseguir lo que nos vamos proponiendo.

A mis suegros que con esa sabiduría nos guían personal, familiar y empresarialmente.

A mis profesores de maestría que me han dejado con muchas preguntas y que debo iniciar con mi propio aprendizaje, exploración y aplicación.

A todos aquellos con quienes tuve la oportunidad de conocerlos a nivel profesional y educativo, que compartieron sus experiencias y conocimientos, ampliaron mi visión del mundo profesional, que, con sus historias y logros, demostraron que la excelencia no tiene fronteras. Su influencia ha sido mi motivación constante para alcanzar lo inexplorado.

Ing. Cristina Elizabeth Pavón Domínguez

AGRADECIMIENTOS

Agradezco primeramente a Dios, a Jesús del Gran Poder, al Divino Niño, La Santísima Virgen María y al Hermano Miguel por concederme la oportunidad de finalizar satisfactoriamente esta nueva etapa universitaria.

A la Universidad Internacional del Ecuador y a EIG Business School, por los conocimientos obtenidos durante este masterado, los cuales desempeñarán un papel crucial en mi trayectoria, tanto a nivel laboral como en mi desarrollo personal.

A mis tías Carmen, Lalita, Silvia y Nelly por darme todo su apoyo y ser un soporte en mi vida para poder seguir adelante.

A mis amigas que conformaron este proyecto: Jacque, Cris y Dome, por unir esfuerzos para sacar adelante el proyecto.

Ing. Darío Hernán Herrera Herrera

Quiero expresar mi profundo agradecimiento a Dios, por permitirme alcanzar esta nueva etapa.

A mi madre y hermano, les agradezco por su amor infinito y constante apoyo. Su presencia en los momentos buenos y malos ha sido mi mayor fortaleza.

A mi esposo, mi compañero de vida, agradezco por ser mi apoyo incondicional. Tu amor y aliento han sido mi impulso para superar desafíos y alcanzar mis metas.

A todos los docentes de la Universidad Internacional del Ecuador y a EIG, mi sincero agradecimiento por impartir conocimientos que han enriquecido mi crecimiento profesional.

A Darío, Doménica y Cristina, agradezco por su amistad y por contribuir con sus valiosos conocimientos al desarrollo de este proyecto.

Ing. Jacqueline Vanessa Quispe Tirado

Deseo expresar mi sincero agradecimiento a aquellos cuya influencia ha dejado una huella única en este viaje académico.

A la Universidad Internacional del Ecuador y a EIG Business School, por proporcióname los recursos y el entorno propicio para obtener mayores conocimientos en Big Data durante este masterado.

A los grandes profesionales con quienes día a día trabajamos en este gran proyecto y, que ahora los considero mis amigos: Dari, Jacque y Dome.

Ing. Cristina Elizabeth Pavón Domínguez

A Dios y a la Virgen del Perpetuo Socorro por guiarme y acompañarme en cada momento de mi vida, porque siempre siento su infinito amor y protección en cada paso que doy.

A mis padres, por su amor y su guía, su esfuerzo incansable, su constante apoyo y confianza en mí. Por ser mi mayor ejemplo de corrección, nobleza, tenacidad, y responsabilidad. Son mi motor para seguir superándome constantemente y mi motivación más grande, espero algún día poder retribuir todo lo que con tanto esfuerzo y dedicación nos han dado.

A mis viejitas, por ser ese ejemplo de mujeres intachables y luchadoras, figuras maternas llenas de sabiduría, paciencia y dulzura, que me han acompañado a lo largo de toda mi vida, protegiéndome y guiándome con su ejemplo y sus palabras.

A mi hermano por su inquebrantable fe en mí y su apoyo permanente, aspiro que nuestro deseo de crecer y superarnos permanezca siempre y cumplamos ese sueño juntos; y a toda mi familia, todos y cada uno han contribuido al logro de esta meta.

A mi amigo hermano que la vida me dio la oportunidad de conocer, y que gracias a su soporte incondicional hemos logrado superar cada obstáculo en el camino. Gracias por siempre estar en los buenos y malos momentos.

A mi mejor amigo y compañero de vida, gracias por darme fuerza en mis momentos difíciles, por tus palabras de motivación y apoyo, por tanta paciencia y comprensión, por todo.

A los docentes de esta maestría, quienes contribuyeron a mi preparación profesional y humana.

A los excelentes profesionales con los que tuve la oportunidad de colaborar para la realización de este proyecto, y con los cuales hemos podido crear una bonita amistad: Jacque, Cris y Dari.

Anhelo que este logro me permita acercarme un poco más a la realización de mis sueños, con la bendición de Dios.

Ing. María Doménica Gómez Sánchez

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR i
CERTIFICACIÓN DE AUTORÍA ii
ACUERDO DE CONFIDENCIALIDAD iii
DEDICATORIAS iv
AGRADECIMIENTOS vi
ÍNDICE GENERAL ix
INDICE DE TABLAS xiii
INDICE DE FIGURAS xiv
RESUMEN xviii
ABSTRACT xix
CAPITULO I INTRODUCCIÓN 1
1.1 Definición del problema 1
1.2 Justificación del proyecto 1
1.3 Objetivos del proyecto 2
1.3.1 Objetivo General 2
1.3.2 Objetivos específicos 2
1.4 Alcance del proyecto 2
CAPITULO II MARCO TEÓRICO 4
2.1 Antecedentes 4
2.2 Inteligencia Artificial 4
2.2.1 Clasificación de la Inteligencia Artificial 5
2.3 Procesamiento del Lenguaje Natural (NLP) 6

2.3.1 Tareas del Procesamiento del Lenguaje Natural (NLP) 7
2.3.2 Aplicaciones del procesamiento de lenguaje natural (NLP) en texto 9
2.3.3 Modelos utilizados para el procesamiento de lenguaje natural (NLP) 10
2.4 Bases de datos 16
2.4.1 Bases de datos relacionales (SQL) 16
2.4.2 Bases de datos no relacional (NoSQL) 21
2.5 Lenguajes de programación: 26
2.5.1 Python 27
2.5.2 Visual Studio Code 38
2.6 Técnicas de procesamiento de datos 40
2.6.1 Pre procesamiento de datos 42
2.7 Metodología ágil 44
2.7.1 Tipos de metodologías 45
2.8 Protección de datos personales 51
2.8.1 Aspectos regulatorios según normativa internacional 51
2.8.2 Aspectos regulatorios según normativa local 57
2.9 Análisis Pestel 60
CAPITULO III METODOLOGÍA 62
3.1 Fuentes de información 62
3.1.1 Fuentes internas 62
3.1.2 Fuentes externas 62
3.2 Estructura base de datos relacional 63
3.2.1 Estructura base de médicos 64
3.2.2 Estructura Base de Especialidades 64

3.2.3 Estructura Base de Planes 65
3.2.4 Estructura Base de Visitadores 65
3.2.5 Estructura Base de Supervisores 66
3.2.6 Estructura Base de Planes-Visitadores 66
3.2.7 Estructura Tabla Planes-Visitadores-Médicos 66
3.3 Diccionario de datos 67
3.4 Diagrama entidad-relación 68
3.5 Arquitectura del proyecto 68
3.6 Flujo del proceso del proyecto 69
3.7 Procesos de transformación de datos 70
3.8 Modelos y algoritmos 71
3.9 Análisis PESTEL 72
3.9.1 Ámbito político 72
3.9.2 Ámbito económico 73
3.9.3 Ámbito social 73
3.9.4 Ámbito tecnológico 73
3.9.5 Ámbito ecológico 74
3.9.6 Ámbito legal 75
3.10 Planteamiento Agile 76
CAPITULO IV DESARROLLO 79
4.1 Archivo de recopilación de datos 79
4.2 Implementación de ambiente de desarrollo y test 85
4.3 Programación del proyecto 87
4.3.1 Importación De Bibliotecas 87

4.3.2 Comando ejecutado para la conexión con la BDD de SQL SERVER 87
4.3.3 Carga Del Archivo Excel 89
4.3.4 Pre procesamiento de los datos 90
4.3.5 Modelo de detección de duplicados 95
4.3.6 Conexión y almacenamiento de registros históricos en MongoDB 103
4.3.7 Carga de médicos nuevos a la base de datos SQL Server 105
CAPITULO V RESULTADOS 107
5.1 Impacto de negocio 107
5.2 Indicadores a alcanzar 107
5.2.1 Porcentaje de reducción en el tiempo dedicado a la gestión de datos: 107
5.2.2 Porcentaje de reducción de duplicados 108
5.2.3 Porcentaje de disminución de errores en digitación 109
5.2.4 Presupuesto perdido asignado al médico con relación a incentivos 110
CAPITULO VI CONCLUSIONES Y RECOMENDACIONES 111
6.1 Conclusiones 111
6.2 Recomendaciones 112
REFERENCIAS BIBLIOGRÁFICAS 113
APÉNDICES 115
7.1 Cronograma y planificación 115
7.2 Planificación de recursos 116
7.3 Documentación técnica del proyecto 117
GLOSARIO 118

INDICE DE TABLAS

Tabla 1 *Fases Scrum* 49

Tabla 2 Listado de tablas de la base de datos SQL 63

Tabla 3 *Estructura Base de médicos* 64

Tabla 4 *Estructura Base de especialidades* 64

Tabla 5 *Estructura de la Base de Planes* 65

Tabla 6 *Estructura Base de visitadores* 65

Tabla 7 *Estructura Tabla de Supervisor* 66

Tabla 8 *Estructura Base de Planes-Visitador* 66

Tabla 9 *Estructura Base de Planes-Visitador-Médicos* 67

Tabla 10 *Diccionario de datos* 67

Tabla 11 *Diagrama entidad-relación de las bases de datos* 68

Tabla 12 *Arquitectura del proyecto* 68

Tabla 13: *Flujograma del proceso* 69

Tabla 14 *Metodología Scrum* 78

Tabla 15 *Cálculo en horas y valores de la gestión de datos* 108

Tabla 16 *Cálculo de reducción de médicos duplicados* 108

Tabla 17 *Cuantificación de tipo de errores en digitación* 109

Tabla 18 *Presupuesto perdido por duplicidad en la información* 110

Tabla 19 *Cronograma y planificación* 115

Tabla 20 *Planificación de recursos* 116

INDICE DE FIGURAS

Figura 1 *Cálculo de la distancia de Levenshtein* 12

Figura 2 *Estructura Básica de su sistema de administración de base de datos relacional.* 17

Figura 3 *Tabla de base de datos relacional* 18

Figura 4 *Interfaz gráfica de Microsoft SQL Server Management Studio* 20

Figura 5 *Estructura básica de un sistema de administración de base de datos NoSQL* 22

Figura 6 *Bases de datos No SQL* 24

Figura 7 *Flujo Scrum para un sprint* 48

Figura 8 *Desarrollo de Scrum* 77

Figura 9 *Archivo de recopilación datos* 79

Figura 10 *Código Visual Basic validación apellidos y nombres* 80

Figura 11 *Código Visual Basic validación cédula* 80

Figura 12 *Código Visual Basic validación fecha de nacimiento* 80

Figura 13 *Código Visual Basic validación celular* 81

Figura 14 *Código Visual Basic validación categoría y contactos(numéricos)* 81

Figura 15 *Código Visual Basic validación con catálogo de especialidad* 82

Figura 16 *Código Visual Basic validación con catálogo de zona, subzona, visitador* 82

Figura 17 *Código Visual Basic validación con catálogo supervisor* 83

Figura 18 *Código Visual Basic validación con catálogo de plan* 83

Figura 19 *Código Visual Basic validación de texto* 84

Figura 20 *Código Visual Basic validación campos de fecha y numéricos* 84

Figura 21 *Ambiente de desarrollo y test SQL Server* 85

Figura 22 *Servidor Mongo DB históricos* 85

Figura 23 *Ambiente de consultas Mongo DB* 86
Figura 24 *Programación en VSC* 86
Figura 25 *Importación de bibliotecas* 87
Figura 26 *Conexión BBD SQL Server* 88
Figura 27 *Query base médicos* 88
Figura 28 *Query base de especialidades* 88
Figura 29 *Código para carga de archivo Excel* 89
Figura 30 *Código para seleccionar las columnas a trabajar* 89
Figura 31 *Código para concatenar los campos* 89
Figura 32 *Merge con base de especialidades* 89
Figura 33 *Visualización del archivo cargado* 90
Figura 34 *Código de visualización de tipo de datos del archivo excel* 90
Figura 35 *Código para eliminación de tildes y Ñ* 91
Figura 36 *Código para eliminación de espacios en blanco* 91
Figura 37 *Código para eliminación de filas duplicadas* 91
Figura 38 *Código para conversión a tipo texto* 92
Figura 39 *Código para conversión a tipo texto* 92
Figura 40 *Código para eliminación de caracteres especiales* 93
Figura 41 *Código para conversión a mayúsculas* 93
Figura 42 *Código para conversión a fecha* 94
Figura 43 *Descripción de algoritmo verificador de cédulas* 94
Figura 44 *Código para validación de cédula de identidad* 95
Figura 45 *Código para creas listas* 96
Figura 46 *Código para iteración a través de las filas* 96

Figura 47 *Código para identificación mejor coincidencia* 97
Figura 48 *Código para almacenamiento de coincidencias* 97
Figura 49 *Código para creación de dataframe* 97
Figura 50 *Código para asignación de etiquetas* 98
Figura 51 *Validación cédula base vs archivo* 98
Figura 52 *Validación de especialidad base vs archivo* 98
Figura 53 *Código para cálculo de similitud fuzzy-simple ratio* 98
Figura 54 *Código para visualización de resultados fuzzy-simple ratio* 99
Figura 55 *Código para cálculo de similitud fuzzy-partial ratio* 99
Figura 56 *Código para visualización de resultados fuzzy-partial ratio* 99
Figura 57 *Código para cálculo de similitud fuzzy-token sort ratio* 100
Figura 58 *Código para visualización de resultados fuzzy-token sort ratio* 100
Figura 59 *Código para cálculo de similitud fuzzy-token set ratio* 100
Figura 60 *Código para visualización de resultados fuzzy-token set ratio* 100
Figura 61 *Código para carga del modelo Berth y tokenizador* 101
Figura 62 *Código para tokenizar todas las filas del Excel y base de médicos* 101
Figura 63 *Código para selección de médico con mayor similitud* 102
Figura 64 *Código para visualización de resultados* 102
Figura 65 *Visualización de resultados Modelo BERT* 103
Figura 66 *Conexión y acceso a la base Mongo DB* 103
Figura 67 *Conversión en diccionario de datos* 104
Figura 68 *Código para obtener la fecha de ejecución* 104
Figura 69 *Código para acceder a la colección históricos* 104
Figura 70 *Código para insertar los registros* 104

Figura 71 *Código para filtrar información de médicos nuevos* 105

Figura 72 *Código para establecer conexión con la base SQL Server* 105

Figura 73 *Código para validar la información a cargar* 106

Figura 74 *Query para inserción de datos* 106

Figura 75 *Presupuesto asignado a la gestión de datos* 108

Figura 76 *Número de médicos duplicados* 109

Figura 77 *Tipos de Errores en digitación* 109

Figura 78 *Variación de presupuesto de incentivos* 110

RESUMEN

El propósito de este proyecto es optimizar el proceso de detección de datos duplicados y enriquecer la calidad de la información en las bases de datos de un Laboratorio Farmacéutico. Se propone el diseño de un modelo de lenguaje natural que verifique automáticamente la existencia de duplicados, lo que contribuirá a reducir la acumulación de información errónea, asignaciones repetidas de presupuestos y gastos operativos, así como a acelerar el procesamiento y verificación de datos. El trabajo se organiza en seis capítulos. El primero contextualiza el problema, presenta la justificación, los objetivos y el alcance del proyecto. El segundo capítulo proporciona un marco teórico detallado que abarca los conceptos clave de NLP, inteligencia artificial y bases de datos. En el tercer capítulo se detallan los algoritmos y técnicas avanzadas de inteligencia artificial utilizados. El cuarto capítulo aborda el desarrollo y los resultados obtenidos, destacando que el Modelo de Fuzzywuzzy - token sort ratio demostró ser eficaz en la detección exitosa de datos médicos nuevos y duplicados. En el quinto capítulo refleja los resultados esperados tras la implementación del modelo, evidenciando mejoras significativas en la precisión y eficiencia de detección de duplicados, así como en la optimización de recursos y tiempos de procesamiento. El último capítulo concluye que la implementación de este modelo no solo incrementó la eficiencia operativa y redujo los riesgos y costos asociados con datos duplicados, sino que también, mejoró considerablemente la experiencia del usuario al proporcionar resultados más precisos y relevantes. La adopción de este modelo de NLP refleja la disposición de la empresa para mantenerse a la vanguardia de la evolución tecnológica, asegurando una gestión de datos más efectiva y una toma de decisiones informada en un entorno empresarial dinámico.

ABSTRACT

The purpose of this project is to optimize the process of duplicate data detection and to enrich the quality of information in the databases of a Pharmaceutical Laboratory. It is proposed the design of a natural language model that automatically verifies the existence of duplicates, which will contribute to reduce the accumulation of erroneous information, repeated budget allocations and operational expenses, as well as to accelerate the processing and verification of data. The paper is organized into six chapters. The first chapter contextualizes the problem, presents the justification, objectives and scope of the project. The second chapter provides a detailed theoretical framework covering the key concepts of NLP, artificial intelligence and databases. The third chapter details the advanced artificial intelligence algorithms and techniques used. The fourth chapter discusses the development and results obtained, highlighting that the Fuzzywuzzy - token sort ratio model proved to be effective in the successful detection of new and duplicate medical data. The fifth chapter reflects the expected results after the implementation of the model, showing significant improvements in the accuracy and efficiency of duplicate detection, as well as in the optimization of resources and processing times. The last chapter concludes that the implementation of this model not only increased operational efficiency and reduced the risks and costs associated with duplicate data, but also, significantly improved the user experience by providing more accurate and relevant results. The adoption of this NLP model reflects the company's willingness to stay at the forefront of technological evolution, ensuring more effective data management and informed decision making in a dynamic business environment.

PALABRAS CLAVES: NLP; Datos; Duplicidad; Optimización; calidad; Eficiencia; IA.

CAPITULO I INTRODUCCIÓN

1.1 Definición del problema

El Laboratorio Farmacéutico maneja una gran cantidad de datos, tales como: datos de los médicos, nombres de los visitadores, nombre de productos, entre otros. En la actualidad existen procesos manuales dentro de la Compañía, que implica la ocurrencia de errores humanos en la digitación y duplicación de información, lo que dificulta el posterior análisis de datos. Por tal motivo, existe la necesidad de identificar y eliminar dichos datos.

La detección manual de duplicados en grandes conjuntos de datos es un proceso laborioso y propenso a errores, lo que hace indispensable el desarrollo de un modelo de procesamiento del lenguaje natural (NLP) que permita identificar automáticamente duplicados con precisión y eficiencia.

1.2 Justificación del proyecto

La implementación de un modelo de lenguaje natural que valide o identifique automáticamente los datos duplicados ayudará a la compañía a disminuir la acumulación de información errónea en las diferentes bases de datos, reducción de asignaciones duplicadas de presupuestos, disminución de los gastos operativos, reducción de tiempos en los procesos de tratamiento de datos y verificación de información. También permitirá el análisis adecuando de la información en los procesos posteriores, tales como: segmentación de médicos, cálculo de comisiones, entre otros.

La implementación de validaciones programadas en Excel, como el uso de macros, acompañada de un modelo NLP robusto, permitirá a la organización procesar y analizar los datos de manera más eficiente. Al automatizar la detección de duplicados y la corrección de

errores lingüísticos y semánticos, se reducirá significativamente el tiempo y el esfuerzo requeridos para mantener la integridad de los datos. Esto no solo mejorará la calidad y confiabilidad de la información, sino que también optimizará la toma de decisiones y la gestión de los recursos de la empresa.

1.3 Objetivos del proyecto

1.3.1 Objetivo General

Diseñar un modelo de Procesamiento del Lenguaje Natural para la detección automática de datos duplicados en un Laboratorio Farmacéutico radicado en la ciudad de Quito.

1.3.2 Objetivos específicos

- Recopilar las necesidades específicas del Laboratorio Farmacéutico en relación con la detección de datos duplicados.
- Definir las herramientas de software necesarias para el desarrollo del proyecto propuesto.
- Elaborar el algoritmo y modelo de Procesamiento de Lenguaje Natural (NLP) adecuado para la detección de datos duplicados.

1.4 Alcance del proyecto

Dentro del alcance del proyecto de titulación se ha considerado lo siguiente:

Detectar casos de datos duplicados de los médicos con la mayor precisión y en el menor tiempo posible, logrando así, mejorar la calidad de información almacenada en las bases de datos.

La recopilación de datos será a través de archivos Excel, los cuales son manejados por los supervisores, ellos a su vez consolidan la información recolectada por los visitadores relacionada a la información personal de los médicos, posterior a ello asignan a los visitadores o

representantes y planes con los cuales se realizarán las respectivas visitas. Este archivo Excel tendrá en su estructura una programación de Visual Basic para reducir problemas de digitación mediante catálogos.

Para el proceso de transformación de datos se utilizará un ETL, en donde la información enviada en Excel por parte de los supervisores será ingresada en un proceso de limpieza y calidad de datos, tales como eliminación de espacios en blanco al inicio y al final de los campos nombres y apellidos de los médicos, eliminación de caracteres especiales en todos los campos excepto en el correo, debido a la integridad de las bases de datos se eliminarán las ñ y las tildes, adicional se definirá si la cédula del médico es válida mediante un algoritmo verificar de identificaciones. Posterior a ello se procederá con el modelo de NLP en donde el umbral de identificación de médicos nuevos o antiguos será definido por un porcentaje de similitud mayor al 80% se considerará médico en base, caso contrario médico nuevo.

Como parte del proyecto se considerará el uso de técnicas avanzadas de NLP, como modelos de lenguaje como BERT y fuzzywuzzy para mejorar la precisión y el rendimiento del modelo de detección de duplicados con entornos colaborativos en lenguaje Python como es el caso de Júpiter Notebook.

Como stakeholders en este proyecto tenemos a la Gerencia General, Gerencia Ventas, Gerencia de Marketing. El área de Infraestructura y Tecnología (IT), debido el impacto que pueda tener analítica avanzada y las necesidades de arquitectura.

CAPITULO II MARCO TEÓRICO

2.1 Antecedentes

Este proyecto propone diseñar un modelo de detección de datos duplicados mediante el uso del Procesamiento del Lenguaje Natural (NLP) para optimizar la eficiencia en la gestión de datos, considerando que el Laboratorio Farmacéutico maneja una gran cantidad de datos que son actualizados diariamente a través por los visitadores médicos, conlleva que los reportes del personal mencionado sean digitados manualmente y, en la consolidación de la información proporcionada existe gran frecuencia de; errores de digitación, duplicidad, omisión, confusión. Los conocimientos adquiridos durante la maestría han permitido identificar las herramientas tecnológicas que permiten automatizar el proceso de detección de duplicados, reducir el tiempo y el esfuerzo requerido para mantener la integridad de los datos, ayudando a garantizar la confiabilidad y calidad de la información. Como punto de partida será evaluar las herramientas tecnológicas actuales utilizadas por la Compañía y definir una mejora de estas. Continuamos como segundo paso, evaluar las herramientas digitales y técnicas de NLP, que permitan proponer un adecuado modelo de datos.

2.2 Inteligencia Artificial

La inteligencia artificial (IA) es un subcampo de la ciencia de la información que tiene como objetivo realizar actividades que normalmente requieren inteligencia humana a través de máquinas inteligentes, como el procesamiento visual, el reconocimiento de voz, la toma de decisiones y el procesamiento del lenguaje. Se utiliza en diferentes industrias, desde la atención médica hasta el comercio minorista, y es responsable de impulsar una amplia gama de avances tecnológicos.

La IA se emplea en una variedad de aplicaciones, incluido el procesamiento del lenguaje natural (NLP), la automatización, la visión por computadora y el reconocimiento de patrones. Se puede utilizar para realizar procesos rutinarios, aumentar la precisión analítica y predictiva y proporcionar información basada en datos. Ésta tiene el potencial de revolucionar muchas industrias y hacer nuestras vidas más fáciles, seguras y eficientes (Hemachandran K., 2023).

2.2.1 Clasificación de la Inteligencia Artificial

La IA ahora consta de muchos subcampos y utiliza una variedad de técnicas, entre las principales que se pueden mencionar tenemos:

1. **Procesamiento del habla**: comprender el habla, la generación de voz, el diálogo automático.
2. **Ingeniería y Sistemas Expertos**: Resolución de problemas de diagnóstico médico, sistemas de soporte de decisiones, sistemas de enseñanza.
3. **Redes Neuronales y Visión Artificial**: Algoritmos genéticos, Modelado cerebral, predicción de series temporales, clasificación, reconocimiento de objetos, comprensión de imágenes, control inteligente, exploración autónoma.
4. **Procesamiento del lenguaje natural (NLP):** El procesamiento del lenguaje natural (NLP) es una colección de técnicas computacionales para el análisis automático y representación de los lenguajes humanos. Sin embargo, el análisis automático de texto, al igual que para los seres humanos, requiere una comprensión mucho más profunda del lenguaje natural por parte de las máquinas. El NLP comprende recuperación de información, traducción automática, preguntas/respuestas, resúmenes, entre otros. (Chowdhary, 2020)
5. **Aprendizaje Automático (ML):** También conocido como análisis predictivo o aprendizaje estadístico es una rama de la inteligencia artificial que permite a las computadoras aprender

y mejorar por su cuenta con o sin datos de entrenamiento, se encuentra en la intersección del campo de investigación de la informática y estadística y se centra en la creación de programas informáticos capaces de acceder y comprender datos con el propósito de extraer conocimiento de estos (Hemachandran K., 2023).

a) **Aprendizaje supervisado**: este método de aprendizaje permite al modelo alimentarse de datos etiquetados. Los datos se utilizan para enseñar a Algoritmo sobre cómo predecir el resultado dado un conjunto de entradas.
b) **Aprendizaje no supervisado**: este método de aprendizaje incluye el ingreso de datos sin etiquetar al sistema. El programa puede entonces reconocer patrones en la información para pronosticar los resultados basados en los mismos (Guido, 2017).

2.3 Procesamiento del Lenguaje Natural (NLP)

El procesamiento del lenguaje natural (NLP) hace referencia a la rama de la informática (y más específicamente, a la rama de la inteligencia artificial o IA encargada de dar a los ordenadores la capacidad de comprender textos y palabras habladas de la misma manera que los seres humanos. (IBM, 2023)

NLP combina la lingüística computacional (modelado basado en reglas del lenguaje humano) con modelos estadísticos, de machine learning y deep learning. Juntas, estas tecnologías permiten a los ordenadores procesar el lenguaje humano en forma de datos de texto o voz y "comprender" su significado completo, junto con la intención y el sentimiento del orador o escritor (IBM, 2023).

NLP impulsa programas que traducen de un idioma a otro, responden a órdenes habladas y resumen grandes volúmenes de texto rápidamente, incluso en tiempo real. Es muy probable que haya interactuado con NLP en forma de sistemas GPS operados por voz, asistentes digitales,

software de dictado de voz a texto, chatbots de servicio al cliente y otros servicios para el consumidor. Sin embargo, NLP también juega un papel cada vez mayor en las soluciones empresariales que permiten optimizar las operaciones de negocio, aumentar productividad de los empleados y simplificar los procesos de negocio de misión crítica (IBM, 2023).

2.3.1 Tareas del Procesamiento del Lenguaje Natural (NLP)

El lenguaje humano está lleno de ambigüedades que hacen increíblemente difícil escribir software que determine con precisión el significado deseado de los datos de texto o voz. Los homónimos, los homófonos, el sarcasmo, las expresiones idiomáticas, las metáforas, las excepciones de gramática y uso o las variaciones en la estructura de la oración son solo algunas de las irregularidades del lenguaje humano que los humanos tardan años en aprender, pero que los programadores deben enseñar a reconocer y entender con precisión desde el principio a las aplicaciones basadas en el lenguaje natural si quieren ser útiles (IBM, 2023). Varias tareas de NLP desglosan los datos de voz y texto humanos de manera que el sistema pueda dar sentido a lo que está ingiriendo. Algunas de estas tareas son (IBM, 2023):

- **El reconocimiento de voz**, también denominado software de voz a texto es la tarea de convertir de manera fiable los datos de voz en datos de texto. El reconocimiento de voz es necesario para cualquier aplicación que siga órdenes de voz o que responda a preguntas habladas. Lo que hace que el reconocimiento de voz sea especialmente difícil es la forma en la que hablan las personas: rápidamente, arrastrando las palabras, con énfasis y entonación variables, en diferentes acentos y, a menudo, usando una gramática incorrecta (IBM, 2023).

- **El etiquetado de parte del discurso**, también denominado etiquetado gramatical, es el proceso de determinar la parte del discurso de una palabra o fragmento de texto específico con base en su uso y contexto. La parte del discurso identifica "lógica" como sustantivo en "La lógica de la frase" y como adjetivo en "La frase es lógica" (IBM, 2023).
- **La desambiguación del sentido de la palabra** es la selección del significado de una palabra con varios significados a través de un proceso de análisis semántico que determina la palabra que tiene más sentido en cada contexto. Por ejemplo, la desambiguación del sentido de la palabra permite distinguir el significado del sustantivo "vaca" en "la vaca del coche" (objeto) y en "vaca que ríe" (animal) (IBM, 2023).
- **El reconocimiento de entidad denominada**, o NEM, identifica palabras o frases como entidades útiles. NEM identifica "Valencia" como una ubicación o "Alfredo" como el nombre de un hombre (IBM, 2023).
- **La resolución de correferencia** es la tarea de identificar si y cuando dos palabras se refieren a la misma entidad. El ejemplo más común es determinar la persona u objeto al que se refiere un determinado pronombre (p. ej., "ella" = "María"), pero también puede implicar identificar una metáfora o una expresión idiomática en el texto (p. ej., una frase en la que "oso" no es un animal sino un persona gruesa y peluda) (IBM, 2023).
- El **análisis de opinión** intenta de extraer cualidades subjetivas (actitudes, emociones, sarcasmo, confusión, sospecha) del texto (IBM, 2023).
- La **generación del lenguaje natural** a veces se describe como lo contrario al reconocimiento de voz o el software de voz a texto; es la tarea de convertir información estructurada en lenguaje humano (IBM, 2023).

2.3.2 Aplicaciones del procesamiento de lenguaje natural (NLP) en texto

El Procesamiento del Lenguaje Natural se aplica en una amplia variedad de campos, incluyendo la atención médica, el análisis financiero, la educación, el marketing, la atención al cliente y muchas otras áreas donde el procesamiento de texto desempeña un papel crucial en la toma de decisiones y la automatización de tareas.

A continuación, detallamos aplicaciones NLP en la resolución de problemas de procesamiento de texto:

Clasificación de texto: El NLP se utiliza para clasificar automáticamente el contenido de texto en categorías o etiquetas específicas. Esto es útil en la categorización de correos electrónicos, detección de spam, análisis de sentimientos en redes sociales y mucho más.

Extracción de información: Permite extraer información estructurada de documentos de texto no estructurado. Esto es útil para la extracción de datos de currículums, informes financieros, noticias, etc.

Traducción automática: Las aplicaciones de NLP como Google Translate utilizan modelos de traducción automática para traducir texto entre diferentes idiomas.

Resumen automático: Los sistemas de NLP pueden resumir grandes cantidades de texto en un resumen más breve, lo que es útil en la generación de resúmenes de noticias, documentos largos y otros contenidos.

Generación de texto: Los modelos de lenguaje basados en NLP pueden generar texto humano similar, lo que se aplica en chatbots, asistentes virtuales y la creación automática de contenido.

Análisis de sentimientos: El NLP se usa para determinar la actitud o el sentimiento expresado en un fragmento de texto, lo que es valioso en la monitorización de redes sociales, la

retroalimentación del cliente y la detección de emociones en comentarios de productos. (Pang, 2008)

Extracción de entidades nombradas (NER): El NLP permite identificar y extraer nombres de personas, lugares, organizaciones y otras entidades de un texto. (Jurafsky, 2020)

Búsqueda semántica: El NLP mejora las capacidades de búsqueda en motores de búsqueda y sistemas de recuperación de información, permitiendo búsquedas más precisas y relevantes. (Jurafsky, 2020)

Resolución de preguntas (Question Answering): Los sistemas de NLP pueden responder preguntas específicas formuladas en lenguaje natural, lo que se utiliza en asistentes de voz y motores de búsqueda avanzados. (Jurafsky, 2020)

Detección de plagio: El NLP se utiliza para comparar textos y detectar similitudes que pueden indicar plagio. (Jurafsky, 2020)

Para el desarrollo de este proyecto la búsqueda semántica es apropiada para la búsqueda de duplicados que nos permitirán identificar con mayor precisión.

2.3.3 Modelos utilizados para el procesamiento de lenguaje natural (NLP)

Los seres humanos podemos distinguir con facilidad la intención de una palabra mal escrita, sin embargo, para las computadoras puede que no sea tan clara dicha distinción, por tal motivo para el desarrollo del presente proyecto se ha considerado diferentes modelos, que nos ayude a distinguir si los nombres de los médicos son similares o no, en un menor tiempo y con mayor precisión.

2.3.3.1 Modelo Fuzzywuzzy

En el ámbito del análisis de datos y el procesamiento del lenguaje natural, comparar y hacer coincidir cadenas es una tarea común y crucial. Sin embargo, debido a variaciones en la

ortografía, el orden de las palabras y diferencias menores es posible que la coincidencia exacta de cadenas no siempre produzca resultados precisos. Aquí es donde entran en juego los algoritmos de coincidencia de cadenas difusas. (Medium.com, 2023)

El algoritmo de coincidencia difusa de cadenas busca determinar el grado de cercanía entre dos cadenas diferentes. Esto se descubre utilizando una métrica de distancia conocida como "editar distancia". La distancia de edición determina qué tan cerca están dos cadenas al encontrar el número mínimo de "ediciones" necesarias para transformar una cadena en otra. (Dutta, 2023)

Hay cuatro tipos principales de ediciones:

- Insertar una letra
- Eliminar una letra
- Intercambiar dos letras adyacentes
- Reemplazar una letra por otra (Pykes, 2023)

Combinar las operaciones de edición permite descubrir la lista de posibles cadenas que están a N ediciones de distancia, donde N es el número de operaciones de edición. Existen diferentes variaciones cómo calcular la distancia, entre ellas la distancia de Levenshtein. (Pykes, 2023)

La distancia de Levenshtein

Es una métrica que lleva el nombre de Vladimir Levenshtein, quien la consideró originalmente en 1965 para medir la diferencia entre dos secuencias de palabras. Podemos usarlo para descubrir la cantidad mínima de ediciones que debe realizar para cambiar una secuencia de una palabra a otra. (Pykes, 2023)

Figura 1
Cálculo de la distancia de Levenshtein

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Nota: Obtenido de https://images.datacamp.com/image/upload/v1678462991/Levenshtein_f11367775d.png

Donde

denota 0 cuando $a=b$ y 1 en caso contrario. Es importante tener en cuenta que las filas del mínimo anterior corresponden a una eliminación, una inserción y una sustitución en ese orden. Algunos de los algoritmos más comunes incluidos en FuzzyWuzzy son:

Algortimo Fuzz Ratio

Este algoritmo mide la similitud entre dos cadenas calculando el número mínimo de ediciones de un solo carácter (inserciones, eliminaciones o sustituciones) necesarias para transformar una cadena en la otra. Es útil cuando necesitas comparar dos cadenas y determinar su similitud general. Es eficaz para identificar cadenas similares que pueden tener diferencias menores debido a errores tipográficos o variaciones ortográficas. (Medium.com, 2023)

Algortimo Fuzz Partial Ratio

Es bastante similar al anterior, pero considera solo la subcadena que mejor coincide entre dos cadenas. Calcula la puntuación de similitud en función de la longitud de la subcadena común más larga, en lugar de la longitud de la cadena completa. Este enfoque ayuda a manejar casos en los que una cadena es un subconjunto o prefijo de la otra. El algoritmo es útil cuando desea encontrar la similitud entre dos cadenas, centrándose solo en la subcadena que mejor coincide.

Es particularmente eficaz para identificar coincidencias cuando una cadena es un subconjunto o prefijo de la otra. (Medium.com, 2023)

Algortimo Fuzz Token Sort Ratio

El algoritmo Token Sort Ratio tokeniza ambas cadenas de entrada, ordena los tokens alfabéticamente y calcula la puntuación de similitud en función de la relación Fuzz entre las listas de tokens ordenadas. Maneja casos en los que el orden de las palabras difiere, pero existe el mismo conjunto de palabras en ambas cadenas. Token Sort Ratio es útil cuando desea comparar cadenas y considerar variaciones en el orden de las palabras. Es particularmente eficaz cuando se espera que las palabras sean similares, pero su posición puede diferir. (Medium.com, 2023)

Algortimo Fuzz Token Set Ratio

Tokeniza ambas cadenas de entrada, elimina tokens duplicados y calcula la puntuación de similitud en función de la intersección y unión de los conjuntos de tokens. Capta la esencia del contenido de las cadenas en lugar de su orden específico. Token Set Ratio es útil cuando desea comparar cadenas independientemente del orden de las palabras. Es eficaz para escenarios en los que la disposición de las palabras puede variar pero el contenido general sigue siendo similar. (Medium.com, 2023)

Algunas de las características clave de FuzzyWuzzy incluyen:

- **Cálculo de similitud:** FuzzyWuzzy proporciona diversas funciones para calcular la similitud entre cadenas de texto, lo que es útil en la comparación de registros, corrección de ortografía y de duplicación de datos. (GitHub, 2023)

- **Opciones de tokenización:** FuzzyWuzzy permite personalizar la tokenización y el procesamiento de cadenas para adaptarse a las necesidades específicas del problema. (GitHub, 2023)
- **Selección de la mejor coincidencia:** Puedes utilizar la función fuzz.extractOne para encontrar la mejor coincidencia entre una cadena de consulta y una lista de cadenas. (GitHub, 2023)
- **Escalabilidad:** FuzzyWuzzy es eficiente y escalable, lo que lo hace adecuado para aplicaciones que involucran grandes conjuntos de datos. (GitHub, 2023)
- **Fácil de usar:** La biblioteca es fácil de utilizar y está disponible a través de la instalación con pip en Python. (GitHub, 2023)

2.3.3.2 Modelo Bert

"BERT" (Bidirectional Encoder Representations from Transformers) (Devlin, 2008), que es uno de los modelos de procesamiento de lenguaje natural más influyentes y ampliamente utilizados desarrollado por Google.

Lo que hace que BERT sea particularmente poderoso es su capacidad para comprender el significado de las palabras en el contexto de las palabras que las rodean en una oración. A diferencia de los modelos de procesamiento de lenguaje natural anteriores que procesaban el texto en una sola dirección (de izquierda a derecha o viceversa), BERT es "bidireccional", lo que significa que puede tener en cuenta tanto las palabras anteriores como las posteriores en una oración al procesar cada palabra.

Este enfoque bidireccional ayuda a BERT a capturar mejor el significado contextual de las palabras y a abordar problemas de ambigüedad en el lenguaje natural. BERT ha sido entrenado

en grandes cantidades de datos textuales y ha demostrado ser muy efectivo en tareas como la comprensión de preguntas, la traducción automática y la mejora de los resultados de búsqueda. Algunas de las características clave de BERT incluyen:

- **Bidireccionalidad:** BERT es capaz de capturar el contexto en ambas direcciones en una oración, lo que lo hace más hábil para comprender el significado y la relación entre palabras.
- **Pre-entrenamiento y ajuste fino:** BERT se inicia su entrenamiento al exponerlo inicialmente a extensos conjuntos de texto no supervisados. Esta fase inicial posibilita que el modelo adquiera representaciones de palabras enriquecidas contextualmente. Posteriormente, se lleva a cabo un proceso de ajuste fino utilizando conjuntos de datos más reducidos y específicos para tareas de Procesamiento del Lenguaje Natural (NLP). Este enfoque refinado permite que BERT aplique su comprensión contextual más general a tareas particulares y más especializadas en el ámbito del procesamiento del lenguaje.
- **Transferencia de conocimiento:** Debido a su pre-entrenamiento en una gran cantidad de datos, BERT ha demostrado ser altamente efectivo en una variedad de tareas de NLP, como el etiquetado de entidades, la clasificación de texto, la traducción automática, el resumen de texto y más.
- **Amplia disponibilidad:** BERT y sus variantes, como RoBERTa, GPT-2 y otros, están disponibles como modelos pre-entrenados y se pueden utilizar en diversas aplicaciones a través de bibliotecas de Python como Hugging Face Transformers.
- **Elevado rendimiento:** BERT y sus derivados han logrado un rendimiento líder en muchas tareas de procesamiento de lenguaje natural y han establecido un estándar alto en el campo.

2.4 Bases de datos

Los sistemas de bases de datos son herramientas de software para describir, almacenar y consultar datos de forma independiente de la aplicación. Todos los sistemas de bases de datos contienen un componente de almacenamiento y de gestión. El componente de almacenamiento llamado base de datos incluye todos los datos almacenados de forma organizada. El componente de gestión llamado “Database Management System” (DBMS) contiene un lenguaje de consulta y manipulación de datos para evaluar y editar datos e información. Este componente también administra todos los permisos de acceso y edición para usuarios y aplicaciones.

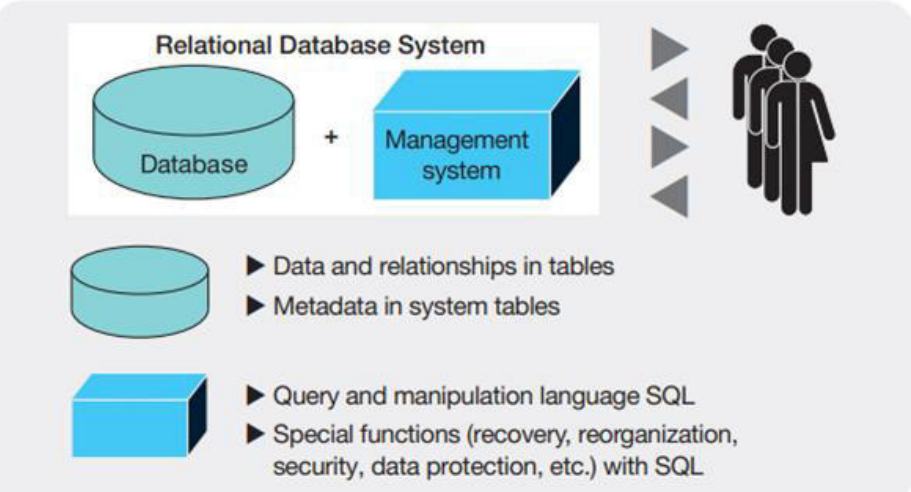
2.4.1 Bases de datos relacionales (SQL)

Las bases de datos se utilizan para almacenar datos de forma centralizada, permanente y estructurada.

Como se muestra en la Figura 2, los sistemas de gestión de bases de datos relacionales son sistemas integrados para la gestión coherente de tablas. Estos ofrecen funcionalidades de servicio y el lenguaje SQL (*Structured Query Languaje*) para la selección y manipulación de datos. Cada sistema de gestión de bases de datos relacionales consta de un componente de almacenamiento y uno de gestión (Kaufmann, 2019)

El componente de almacenamiento es decir la base de datos, guarda los datos y las relaciones entre ellos. Además, contiene los datos del sistema predefinidos necesarios para el funcionamiento de la base de datos. Esta información descriptiva puede ser consultada, pero no manipulada por los usuarios.

Figura 2
Estructura Básica de su sistema de administración de base de datos relacional.



Nota: Obtenido del Libro SQL & NoSQL Databases por Kaufmann, A. Meier y M. (2019)

Las bases de datos relacionales estructuran la información en tablas que pueden interconectarse según a partir de datos similares, lo que facilita la obtención de una tabla completamente nueva que combine los datos de una o más tablas a través de una única consulta.

Adicionalmente brinda al personal una mejor compresión acerca de las relaciones entre todos los datos disponibles, facilitando la toma de decisiones.

Una tabla es un conjunto de filas (registros) y columnas (atributos o características) que deben cumplir con los siguientes requisitos (IBMEducation, 2019):

Nombre de la tabla: Una tabla tiene un nombre de tabla único.

Clave de identificación: Un atributo o una combinación de atributos identifica de forma única los registros dentro de la tabla.

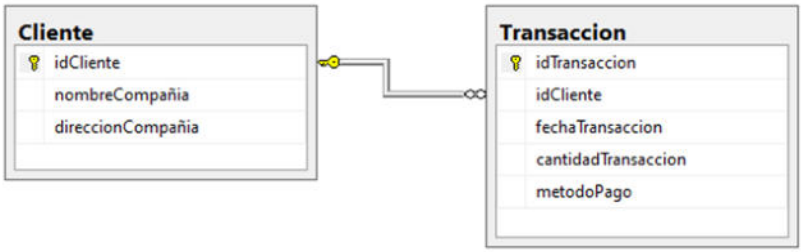
Nombre de atributo: Todos los nombres de atributo son únicos dentro de una tabla y etiquetan una columna específica de la misma con la característica/propiedad requerida.

Por ejemplo, se puede mencionar una empresa la cual dispone de una tabla de clientes que contiene información sobre cada uno de ellos, y una o más tablas de transacciones con información de las transacciones individuales.

Los atributos o propiedades para la tabla de clientes incluirían elementos como el ID de cliente, nombre de la compañía, dirección de la empresa, entre otros. Por otro lado, las columnas para una tabla de transacciones podrían abarcar la fecha de la transacción, el ID de cliente, la cantidad de la transacción, el método de pago, entre otros. Estas tablas se relacionan a través de un atributo clave, en este caso, sería el campo del ID de cliente (ver Figura 2).

En consecuencia, es posible generar informes significativos a partir de consultas generadas a una tabla, como un resumen consolidado de clientes. Los generadores de informes toman estas consultas y las ejecutan según sea necesario para generar informes formales. (IBM, 2023)

Figura 3
Tabla de base de datos relacional



Nota: Obtenido de https://www.ibm.com/mx-es/topics/relational-databases

La parte más importante del componente de gestión es la definición de datos relacionales, selección y lenguaje de manipulación SQL, ya que mediante una consulta se pueden realizar distintas operaciones en la información, como: búsquedas, inserciones, modificaciones, entre otras; el sistema de gestión compila la consulta y devuelve los resultados de esta.

Este sistema también contiene funciones de servicio para la restauración de datos después de errores, para la protección de datos y para la copia de seguridad.

2.4.1.1 Database Management System (DBMS)

Un Sistema de gestión de base de datos es un software para un fácil, eficiente y confiable procesamiento y gestión de datos. Se utiliza para:

- Creación.
- Recuperación de información.
- Actualización.
- Gestionar.

Entre las múltiples funcionalidades que proporciona están: control de redundancia, facilita la gestión de memoria, otorga acceso y permisos para usuarios a la base de datos, capacidad de ampliación, escalabilidad y flexibilidad de los datos (MicrosoftDocumentation, 2018).

Algunos de los sistemas de gestión de base de datos más utilizados son: Oracle, Microsoft SQL Server, Access, etc.

2.4.1.2 SQL Server Management Studio

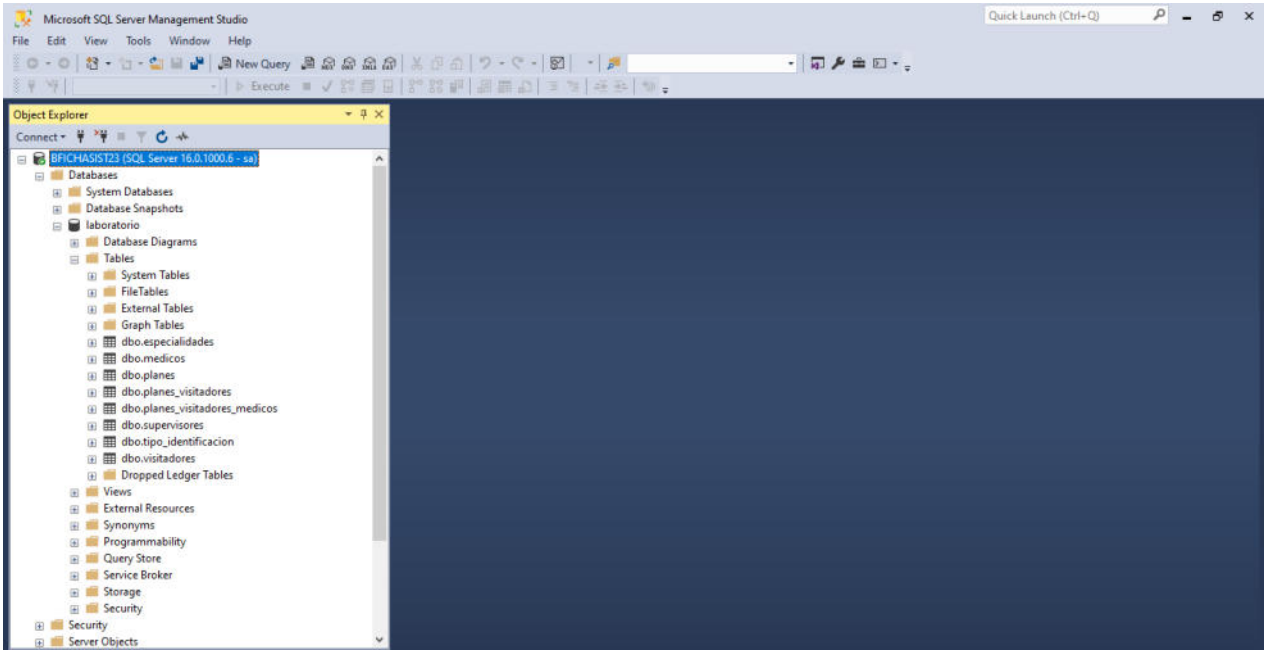
SQL Server Management Studio (SSMS), como su denominación sugiere, sirve como la interfaz principal de administración que conecta al administrador de bases de datos con SQL Server. Este último es el sistema principal de gestión utilizado por Microsoft para bases de datos tradicionales.

SSMS se utiliza para ingresar, configurar, y gestionar todos los componentes de SQL Server. SSMS integra un conjunto de herramientas gráficas con varios editores de código para brindar acceso a desarrolladores y administradores de bases de datos a SQL Server (M. Radivojevic, 2019)

En la Figura 4 se presenta la interfaz gráfica de SSMS. La misma que está compuesta de los siguientes elementos.

- **Explorador de Objetos**: Utilizado para visualizar y gestionar todos los objetos en una o varias instancias de SQL Server. Al ejecutar SSMS se solicita conectarse a una instancia de SQL, posteriormente es posible conectarse a varias de éstas y visualizarlas
- **Editor de Código**: En donde mediante comandos o instrucciones se pueden gestionar los servidores y objetos de las bases de datos.

Figura 4
Interfaz gráfica de Microsoft SQL Server Management Studio



2.4.1.3 Structured Query Language

SQL o Lenguaje de consulta estructurado, es un lenguaje diseñado específicamente para comunicarse con bases de datos.

A diferencia de otros idiomas (como Java o Visual Basic), SQL se compone de muy pocas palabras; esto es deliberado. SQL está diseñado para proporcionar una forma simple y eficiente de leer y escribir datos en una base de datos.

Algunas de las ventajas de SQL son:

- SQL no es un lenguaje propietario utilizado por proveedores de bases de datos específicos. Casi todos los Sistemas de administración de bases de datos principales admiten SQL, por lo que aprender este idioma posibilitará la interacción con casi todas las bases de datos con las que se encontrará.
- SQL es fácil de aprender. Los comandos están formados por palabras descriptivas en inglés, y no hay muchos de ellos.
- SQL es un lenguaje muy poderoso, mediante el uso inteligente de sus elementos de lenguaje puede realizar operaciones de bases de datos muy complejas y sofisticadas (Forta, 2004).

2.4.2 Bases de datos no relacional (NoSQL)

Si bien "no relacional" sería una mejor descripción que NoSQL, este último ha prevalecido entre desarrolladores y distribuidores de bases de datos en el mercado en años recientes.

La expresión NoSQL ahora se ocupa para cualquier manejo de datos con un enfoque no relacional, que satisface los siguientes criterios (Kaufmann, 2019):

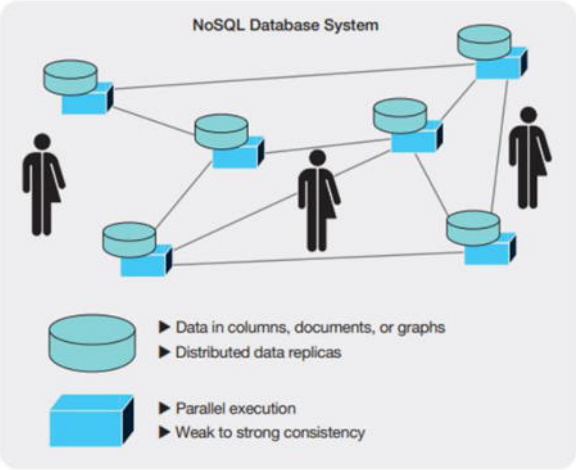
- Tablas no son usadas para almacenar datos.
- SQL no es utilizado como lenguaje.

NoSQL se puede entender como "No solo SQL", un método para el diseño de bases de datos que proporciona esquemas flexibles para almacenar y recuperar información más allá de las

estructuras de tablas tradicionales de las bases de datos relacionales. Cabe recalcar, que la tendencia actual es utilizar las bases de datos relacionales y NoSQL en una sola aplicación. Las bases de datos No SQL se han vuelto más populares en la era de la nube, *big data* y aplicaciones web y móviles de alto volumen. Existe gran cantidad de opciones en cuanto refiere a bases de datos No SQL, es por esto por lo que se selecciona la opción más adecuada teniendo en cuenta criterios como escalabilidad, rendimiento y facilidad de uso (IBMCloudEducation, 2019).

La estructura básica de un sistema de gestión de base de datos NoSQL se muestra en la Figura 5. Los sistemas de gestión de bases de datos NoSQL utilizan principalmente una arquitectura de almacenamiento masivo distribuido. Para garantizar una alta disponibilidad y evitar interrupciones en los sistemas de base de datos NoSQL, varios conceptos de redundancia son comúnmente utilizados.

Figura 5
Estructura básica de un sistema de administración de base de datos NoSQL



Nota: Obtenido de https://www.ibm.com/cloud/learn/nosql-databases.

Los tipos de bases de datos NoSQL más conocidos y utilizados son: Bases de datos de valores clave, documentos, columnas y gráficos.

La Figura 6 muestra tres sistemas diferentes de gestión de bases de datos NoSQL.

Base de datos valores clave: son las más simples. Los datos se almacenan como entidades que contienen una clave de identificación (clave = "clave") y una lista de valores (valor = "valor 1", "valor 2") (Kaufmann, 2019).

Por ejemplo, una tienda en línea con gestión de sesiones y carrito de compras. La ID de sesión es la clave de identificación; Los artículos individuales del carro se almacenan como valores.

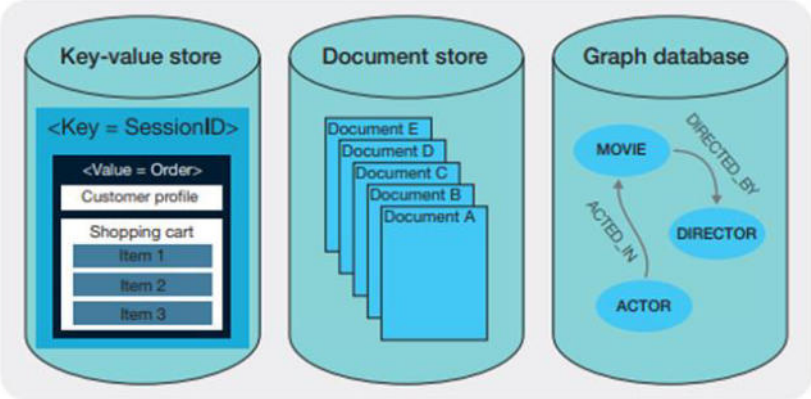
Bases de datos de documentos: Las bases de datos documentales, también conocidas como bases de datos orientadas a documentos, utilizan un método de almacenamiento de información basado en valores clave, aunque con diferencias notables en comparación con las bases de datos de valores clave. Estas bases de datos almacenan información en forma de documentos, que son entidades semiestructuradas comúnmente en un formato estándar como la notación de objetos JavaScript (JSON) o el lenguaje de marcado extensible (XML). Es importante destacar que al emplear la expresión "documento" en este contexto, no se está haciendo referencia a procesadores de texto ni a otros archivos de oficina. En cambio, se hace alusión a esquemas de datos que se guardan como cadenas de caracteres o representaciones de cadenas binarias (Sullivan, 2015).

Bases de datos de columnas: Las bases de datos de columnas, también conocidas como bases de datos de familias de columnas, se destacan por su alta capacidad de escalabilidad. Permiten a los desarrolladores modificar con flexibilidad las columnas dentro de una familia de columnas. Aunque comparten términos similares con las bases de datos tradicionales, como el concepto de columnas, en este contexto, dichas columnas representan un registro, a diferencia

de una base de datos convencional, donde representan un atributo de un registro (Sullivan, 2015).

Bases de datos de gráficos: Las bases de datos de gráficos son las más especializadas de las cuatro bases de datos NoSQL. En lugar de modelar datos usando columnas y filas, una base de datos de gráficos usa estructuras llamadas nodos y relaciones (en discusiones más formales, se llaman vértices y aristas). Un nodo se define como una entidad que posee un conjunto de atributos junto con un identificador único. Por otro lado, una relación se configura como la conexión entre dos nodos, los cuales incluyen atributos relacionados con dicha conexión (Sullivan, 2015).

Figura 6
Bases de datos No SQL



Nota: Obtenido de NoSQL for Mere Mortals por Sullivan Dan (2015)

2.4.2.1 MongoDB

MongoDB es una base de datos NoSQL que puede almacenar datos tanto estructurados como no estructurados. Proporciona funciones que se necesitan para almacenar Big Data del mundo real. Su diseño basado en documentos hace que sea fácil de entender y utilizar. Está diseñado para ser utilizado tanto en aplicaciones experimentales como del mundo real y es más fácil de

configurar y de administrar que la mayoría de las otras bases de datos NoSQL. Además, su sintaxis intuitiva para consultas y comandos hace que sea fácil de aprender (MongoDB, 2023). MongoDB cuenta con las siguientes características:

1. **Esquema flexible y dinámico**: Posee un esquema para su base de datos que permite variaciones en los campos de diferentes documentos. En términos simples, cada registro de la base de datos puede tener o no la misma cantidad de atributos. Aborda la necesidad de almacenar datos sin realizar ningún cambio en el esquema en sí (MongoDB, 2023).
2. **Lenguaje de consulta enriquecido**: Admite un lenguaje de consulta enriquecido e intuitivo, lo que significa consultas simples pero potentes. Viene con un vasto entorno de agregación que le permite agrupar y filtrar datos según sea necesario. También tiene soporte integrado para búsqueda de texto de propósito general y propósitos específicos como búsquedas geoespaciales (MongoDB, 2023).
3. **Transacciones ACID de múltiples documentos**: Atomicidad, Coherencia, integridad y Durabilidad (ACID) son características que permiten que sus datos se almacenen y actualicen para mantener su consistencia de sus transacciones. MongoDB admite ACID en transacciones de un solo documento y de varios documentos (MongoDB, 2023).
4. **Alto rendimiento**: Proporciona un alto rendimiento utilizando modelos de datos integrados para reducir el uso de E/S del disco. Además, el amplio soporte para indexar diferentes tipos de datos para agilizar las consultas. La indexación es un mecanismo para mantener punteros de datos relevantes en un índice, al igual que el índice de un libro (MongoDB, 2023).
5. **Alta disponibilidad**: Admite clústeres distribuidos con un mínimo de tres nodos. Un clúster se refiere a una implementación de base de datos que utiliza múltiples nodos/máquinas para

el almacenamiento y recuperación de datos. Las conmutaciones por error son automáticas y los datos se replican en nodos secundarios de forma asincrónica (MongoDB, 2023)

6. **Escalabilidad**: Proporciona una forma de escalar sus bases de datos horizontalmente en cientos de nodos (Amit Phaltankar, 2020).

2.5 Lenguajes de programación:

Los lenguajes de programación son un conjunto de instrucciones y reglas utilizadas para escribir programas de computadora. Estas instrucciones le dicen a una computadora qué hacer y cómo hacerlo. Los lenguajes de programación son una forma de comunicación entre los seres humanos y las máquinas, permitiendo a los programadores escribir código que las computadoras pueden entender y ejecutar.

Algunos aspectos importantes sobre los lenguajes de programación incluyen:

Sintaxis y Semántica: Cada lenguaje de programación tiene su propia sintaxis (las reglas para escribir código) y semántica (el significado de las instrucciones). Los programadores deben seguir estas reglas para que el código sea válido y funcione correctamente.

Tipos de Lenguajes: Los lenguajes de programación se dividen en varios tipos, como lenguajes de alto nivel (más fáciles de entender para los humanos) y lenguajes de bajo nivel (más cercanos a la máquina y utilizados para programación de sistemas). Ejemplos de lenguajes de alto nivel incluyen Python, Java, C++, y Ruby, mientras que ejemplos de lenguajes de bajo nivel incluyen C y ensamblador.

Compilados e Interpretados: Algunos lenguajes se compilan, lo que significa que el código se traduce completamente a un lenguaje que la máquina puede ejecutar antes de la ejecución. Otros lenguajes se interpretan, lo que significa que el código se traduce línea por línea durante

la ejecución. Ejemplos de lenguajes compilados incluyen C y C++, mientras que Python y JavaScript son ejemplos de lenguajes interpretados.

Propósito Específico: Algunos lenguajes de programación están diseñados para tareas específicas, como el análisis de datos (R), desarrollo web (JavaScript), cálculos científicos (MATLAB) o desarrollo de aplicaciones móviles (Swift).

Bibliotecas y Frameworks: Muchos lenguajes tienen bibliotecas y frameworks que permiten a los programadores utilizar código preexistente para tareas comunes, lo que acelera el desarrollo de aplicaciones.

Los idiomas de codificación representan herramientas esenciales en el ámbito de la informática y la programación, desempeñando un papel clave en diversas aplicaciones, que van desde la creación de software y aplicaciones móviles hasta la ciencia de datos, la inteligencia artificial, la automatización de tareas y más. La selección del lenguaje de programación apropiado se basa en la tarea específica que se desea realizar y en las preferencias individuales del programador.

2.5.1 Python

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo. (Amason Web Services, 2023)

2.5.1.1 Herramientas y bibliotecas Python

Python, un lenguaje de programación polifacético, goza de una extensa popularidad en diversos campos, incluyendo la ciencia de datos, la inteligencia artificial, el desarrollo web, entre otros. Hay una gran cantidad de herramientas y bibliotecas disponibles en Python que facilitan la

programación y el desarrollo en diversos dominios. A continuación, se detalla las herramientas y bibliotecas de Python que se han utilizado para el diseño del modelo de este proyecto:

Biblioteca PYODBC: La biblioteca PYODBC es un módulo de Python que se utiliza para conectar y comunicarse con bases de datos relacionales a través del estándar ODBC (Open Database Connectivity). ODBC es una interfaz que permite a las aplicaciones acceder a diversas bases de datos, como Microsoft SQL Server, Oracle, MySQL, PostgreSQL y más, utilizando un único conjunto de API. PYODBC proporciona una forma sencilla de conectarse a bases de datos y realizar operaciones de lectura y escritura desde Python. Algunas de las características y funcionalidades clave de PYODBC incluyen:

- **Conexión a bases de datos:** PYODBC permite establecer conexiones con bases de datos mediante una cadena de conexión que especifica el controlador ODBC, la dirección del servidor, el nombre de la base de datos y las credenciales de acceso.
- **Ejecución de consultas SQL:** Puedes utilizar PYODBC para ejecutar consultas SQL, incluyendo SELECT, INSERT, UPDATE, DELETE, y otras instrucciones SQL directamente desde Python.
- **Recuperación de datos:** Puedes recuperar datos de consultas SQL y trabajar con ellos en Python como objetos de Python, como listas o diccionarios.
- **Parámetros y consultas parametrizadas:** PYODBC permite la creación de consultas parametrizadas para evitar la inyección de SQL y facilitar la reutilización de consultas con diferentes valores.
- **Transacciones:** Puedes gestionar transacciones en bases de datos, lo que es útil para operaciones que requieren confirmación o reversión de cambios.

- **Soporte para múltiples bases de datos:** PYODBC es compatible con una variedad de bases de datos, lo que te permite trabajar con diferentes sistemas de gestión de bases de datos utilizando una única interfaz en Python.

Biblioteca PYTORCH: PyTorch que es una biblioteca de código abierto ampliamente utilizada para el aprendizaje profundo (deep learning) en Python. PyTorch es especialmente popular en la comunidad de investigación y desarrollo de modelos de aprendizaje automático y aprendizaje profundo debido a su flexibilidad y facilidad de uso. Algunas de las características y funcionalidades clave de PyTorch incluyen:

- **Aprendizaje Profundo (Deep Learning):** PyTorch se ha creado como una biblioteca especializada en la creación de modelos de aprendizaje profundo, proporcionando una variedad extensa de herramientas y funciones para la construcción, entrenamiento y evaluación de redes neuronales artificiales.
- **Interfaz dinámica:** Una de las características distintivas de PyTorch es su gráfico computacional dinámico. A diferencia de algunas otras bibliotecas que utilizan gráficos estáticos, PyTorch permite la construcción de modelos de manera más flexible, lo que facilita la depuración y la experimentación.
- **Módulos y capas predefinidas:** PyTorch incluye una variedad de módulos y capas predefinidos para acelerar el desarrollo de modelos. Estos módulos se pueden utilizar para construir arquitecturas comunes de redes neuronales, como convolucionales, recurrentes y redes totalmente conectadas.
- **Soporte para GPU:** PyTorch se integra bien con GPU, lo que permite el entrenamiento y la inferencia de modelos en hardware acelerado para mejorar el rendimiento.

- **Ecosistema activo:** PyTorch cuenta con una comunidad activa de desarrolladores, lo que significa que constantemente se están desarrollando nuevas extensiones, módulos y herramientas para ampliar su funcionalidad.
- **Integración con otras bibliotecas:** PyTorch se puede integrar con otras bibliotecas populares de aprendizaje profundo, como torchvision para visión por computadora y torchaudio para procesamiento de audio.
- **Aplicaciones en investigación y producción:** PyTorch se utiliza tanto en entornos de investigación académica como en aplicaciones de producción del mundo real, lo que lo hace versátil en una amplia variedad de aplicaciones de aprendizaje profundo.
- **Comunidad y documentación:** PyTorch cuenta con una comunidad activa y una amplia documentación, lo que facilita su aprendizaje y su uso.

Biblioteca Pandas: La biblioteca "Pandas" se destaca como una herramienta robusta y de código abierto en el entorno de Python, siendo ampliamente empleada para analizar y manipular datos estructurados. En particular, se utiliza con frecuencia para procesar datos en formato tabular y de series temporales. Fue creada por Wes McKinney y es una de las herramientas más esenciales en el kit de herramientas de cualquier persona que trabaje con datos en Python (McKinney, 2013). A continuación, se detallan las características y funcionalidades principales de la biblioteca "pandas":

- **Estructuras de Datos:** Pandas presenta dos estructuras de datos fundamentales: las Series y los DataFrames.
- **Series:** Una estructura de una sola dimensión capaz de almacenar datos de diversa índole, semejante a un arreglo o a una columna en una hoja de cálculo.

- **DataFrame:** Una estructura de dos dimensiones que guarda semejanza con una tabla o una hoja de cálculo, en la cual los datos se disponen en filas y columnas.
- **Lectura y Escritura de Datos:** Pandas proporciona capacidades para la lectura y escritura de datos desde y hacia diversos formatos, incluyendo CSV, Excel, SQL, JSON, HTML, entre otros.
- **Manipulación de Datos:** Pandas facilita la limpieza y transformación de datos, incluyendo la eliminación de valores nulos, la selección y filtrado de datos, la agregación, la concatenación, la fusión y la pivoteación de datos.
- **Indexación y Selección:** Pandas permite una amplia variedad de técnicas de indexación y selección de datos, incluyendo la selección de columnas y filas, la indexación basada en etiquetas o ubicaciones, y la indexación booleana.
- **Agrupación y Agregación:** Pandas ofrece herramientas para agrupar datos en función de una o más columnas y realizar operaciones de agregación, como la suma, el promedio o la mediana en cada grupo.
- **Visualización de Datos:** Aunque no es una biblioteca de visualización en sí, Pandas se integra bien con bibliotecas de visualización como Matplotlib y Seaborn para crear gráficos y visualizaciones de datos.
- **Tratamiento de Fechas y Series Temporales:** Pandas proporciona funciones específicas para trabajar con datos de series temporales, lo que facilita el análisis de datos de tiempo y series temporales.
- **Integración con NumPy:** Pandas se integra bien con la biblioteca NumPy, lo que permite el uso de estructuras de datos de Pandas en combinación con cálculos numéricos de NumPy.

- **Facilidad de Uso:** La biblioteca está diseñada para ser fácil de aprender y usar, lo que la hace accesible para principiantes en ciencia de datos y análisis de datos.

Biblioteca Transformers: Es una librería de código abierto ampliamente utilizada en Python para llevar a cabo el procesamiento del lenguaje natural (NLP) y la implementación de modelos de lenguaje avanzados, incluyendo modelos basados en Transformers, como BERT, GPT-2, RoBERTa, y muchos otros. Esta biblioteca ofrece una amplia gama de modelos preentrenados y herramientas para trabajar con ellos. A continuación, se describen algunas de las características clave de la biblioteca "Transformers" (Vasilev, 2019):

- **Modelos preentrenados:** Transformers proporciona acceso a una variedad de modelos de lenguaje preentrenados, que se han entrenado en grandes conjuntos de datos y pueden ser afinados o utilizados directamente para una amplia gama de tareas de procesamiento del lenguaje natural.
- **Tokenizadores:** La biblioteca incluye tokenizadores eficientes para dividir texto en tokens, lo que es esencial para el procesamiento de texto con modelos de lenguaje. Los tokenizadores son específicos para cada modelo y están optimizados para su uso.
- **Fácil acceso a modelos:** Los modelos preentrenados se pueden cargar y utilizar fácilmente desde la biblioteca, lo que facilita la implementación de soluciones de NLP.
- **Soporte para tareas de NLP:** Transformers ofrece modelos previamente entrenados y utilidades para una diversidad de tareas en el procesamiento del lenguaje natural, que abarcan desde la clasificación de texto, generación de texto, traducción automática, extracción de información, resolución de preguntas, entre otras.

- **Soporte multiplataforma:** La biblioteca funciona en diferentes plataformas, incluyendo PyTorch y TensorFlow, lo que permite a los usuarios elegir la plataforma de su elección para desarrollar modelos de lenguaje.
- **Comunidad activa:** Transformers es respaldado por una comunidad activa de desarrolladores y científicos de datos, lo que significa que se están desarrollando continuamente nuevos modelos y recursos.

Biblioteca Scikit-learn: La librería "scikit-learn" (o sklearn) es ampliamente reconocida como una de las herramientas más empleadas para el aprendizaje automático en Python. Ofrece recursos simples y eficaces para llevar a cabo tareas tanto de aprendizaje supervisado como no supervisado, y dispone de utilidades para evaluar y seleccionar modelos. A continuación, se describen algunas de las características clave de scikit-learn (Scikit-learn_machine learning in Python-scikit-learn 1.0.2 documentation, 2023):

- **Amplia variedad de algoritmos:** Scikit-learn incluye una amplia variedad de algoritmos para clasificación, regresión, clustering, reducción de dimensionalidad, detección de anomalías y más. Esto hace que sea una herramienta versátil para una variedad de aplicaciones de aprendizaje automático (Scikit-learn_machine learning in Python-scikit-learn 1.0.2 documentation, 2023).
- **Interfaz consistente:** Scikit-learn ofrece una interfaz de programación de aplicaciones (API) consistente para la mayoría de sus algoritmos, lo que facilita el uso y la comparación de diferentes métodos (Scikit-learn_machine learning in Python-scikit-learn 1.0.2 documentation, 2023).
- **Preprocesamiento de datos:** La biblioteca proporciona herramientas para la limpieza, transformación y preprocesamiento de datos, incluyendo la estandarización, normalización,

codificación de variables categóricas y manejo de valores faltantes (Scikit-learn_machine learning in Python-scikit-learn 1.0.2 documentation, 2023).

- **Evaluación de modelos:** Scikit-learn ofrece métricas y funciones de evaluación que permiten medir el rendimiento de los modelos, como precisión, F1-score, AUC, entre otras.
- **Selección de modelos:** La biblioteca incluye herramientas para realizar validación cruzada y selección de hiperparámetros, lo que ayuda a encontrar el modelo y la configuración óptimos.
- **Facilidad de uso:** Scikit-learn es conocido por su facilidad de uso, lo que lo hace adecuado para principiantes en aprendizaje automático. La documentación es extensa y contiene muchos ejemplos.
- **Integración con otras bibliotecas:** Scikit-learn se integra de manera fluida con otras bibliotecas de Python, como NumPy y pandas, simplificando la manipulación de datos y la incorporación en flujos de trabajo de análisis de datos.
- **Aprendizaje supervisado y no supervisado:** Scikit-learn soporta una variedad de tareas de aprendizaje, incluyendo regresión, clasificación, clustering y reducción de dimensionalidad, entre otras.
- **Licencia de código abierto:** Scikit-learn es de código abierto y está bajo la licencia BSD, lo que significa que es gratuito y se puede utilizar en proyectos comerciales y académicos.

Biblioteca Fuzzywuzzy: es una herramienta de Python que se utiliza para calcular la similitud entre cadenas de texto y realizar coincidencias aproximadas o difusas. Esta biblioteca utiliza la "Distancia de Levenshtein", también conocida como distancia de edición, para calcular la similitud entre dos cadenas de texto al cuantificar el número de operaciones (inserciones, eliminaciones o sustituciones) necesarias para convertir una cadena en la otra.

Fuzzywuzzy es ampliamente utilizado en tareas como la de duplicación de datos, la corrección ortográfica, la búsqueda de coincidencias aproximadas y la clasificación de cadenas de texto en función de su similitud. Algunas de las funciones y características más comunes de Fuzzywuzzy incluyen:

- **Ratio:** Calcula un índice de similitud entre dos cadenas de texto en una escala de 0 a 100, donde 100 significa una coincidencia perfecta. Es útil para encontrar coincidencias aproximadas.
- **Partial Ratio:** Similar a "Ratio", pero se enfoca en encontrar subcadenas coincidentes en lugar de coincidencias completas.
- **Token Sort Ratio:** Este método divide y ordena las palabras en las cadenas antes de calcular la similitud. Útil para manejar cadenas con palabras desordenadas o palabras en diferente orden.
- **Token Set Ratio:** Similar a "Token Sort Ratio", pero también maneja la diferencia en palabras únicas entre las dos cadenas.
- **Process:** Una función que permite realizar comparaciones y encontrar las mejores coincidencias dentro de un conjunto de cadenas.

Biblioteca Unidecode: se utiliza para transliterar (convertir) cadenas de texto Unicode en caracteres legibles por humanos. En esencia, Unidecode convierte caracteres Unicode complejos, como caracteres acentuados, caracteres cirílicos, caracteres chinos, entre otros, en una representación legible en caracteres latinos sin diacríticos. Esto puede ser útil en diversas aplicaciones, como la normalización de texto, la generación de URL legibles por humanos y la búsqueda de texto en bases de datos o sistemas que no admiten caracteres Unicode (Github, 2023).

Biblioteca RE: Se emplea para manejar expresiones regulares, las cuales son patrones de búsqueda utilizados para identificar secuencias de texto dentro de cadenas de caracteres. La librería "re" ofrece una variedad extensa de funciones y métodos que permiten la creación, manipulación y búsqueda de patrones de expresiones regulares en cadenas de texto (McKinney, 2013).

Algunas de las funciones y métodos más comunes de la biblioteca "re" incluyen:

- **Re.compile(pattern):** Compila una expresión regular dada en un objeto de patrón que puede ser reutilizado para buscar y manipular texto.
- **Re.search(pattern, string):** Busca la primera ocurrencia del patrón en la cadena de texto y devuelve un objeto de coincidencia si se encuentra una coincidencia.
- **Re.match(pattern, string):** Busca el patrón al principio de la cadena de texto y devuelve un objeto de coincidencia si se encuentra una coincidencia.
- **Re.findall(pattern, string):** Encuentra todas las coincidencias del patrón en la cadena de texto y devuelve una lista de todas las coincidencias encontradas.
- **Re.sub(pattern, replacement, string):** Reemplaza todas las ocurrencias del patrón en la cadena de texto con una cadena de reemplazo dada.
- **Re.split(pattern, string)**: Divide la cadena de texto en una lista de subcadenas utilizando el patrón como separador.

La biblioteca "re" es extremadamente poderosa y versátil, y se utiliza comúnmente en tareas de procesamiento de texto, búsqueda y validación de datos. Las expresiones regulares pueden ser muy útiles para buscar patrones específicos en grandes cantidades de texto o realizar operaciones de limpieza de datos basadas en patrones.

Biblioteca Datetime: La biblioteca "datetime" es una biblioteca estándar de Python que se utiliza para trabajar con fechas y horas. Esta biblioteca proporciona clases y funciones para manejar objetos de fecha y hora, realizar cálculos de tiempo, formatear y analizar fechas, y trabajar con diferencias de tiempo. Es una herramienta esencial cuando necesitas realizar tareas relacionadas con el tiempo y las fechas en tus programas Python (McKinney, 2013). Algunas de las clases y funciones más comunes dentro de la biblioteca "datetime" incluyen:

- **Datetime.datetime**: Esta clase se utiliza para representar objetos de fecha y hora. Puedes crear instancias de esta clase para representar una fecha y hora específica.
- **Datetime.date**: Representa objetos de fecha sin información de hora.
- **Datetime.time**: Representa objetos de hora sin información de fecha.
- **Datetime.timedelta**: Esta clase se utiliza para representar diferencias de tiempo, como la duración entre dos fechas o horas.
- **Datetime.now():** Devuelve la fecha y hora actual.
- **Datetime.strptime(string, format):** Convierte una cadena en un objeto de fecha y hora utilizando un formato específico.
- **Datetime.strftime(format)**: Convierte un objeto de fecha y hora en una cadena con un formato específico.

La biblioteca "datetime" es muy versátil y es útil en una amplia variedad de aplicaciones, como cálculos de fechas, programación de tareas basadas en el tiempo, análisis de registros de tiempo y más.

Biblioteca Pymongo: se utiliza para interactuar con bases de datos MongoDB. MongoDB es una base de datos NoSQL que almacena datos en formato JSON (BSON) y es ampliamente

utilizado en aplicaciones web y sistemas que requieren almacenamiento de datos flexible y escalabilidad horizontal (PyMongo, 2023).

PyMongo permite a los desarrolladores de Python conectarse a una base de datos MongoDB, realizar operaciones de lectura y escritura, y administrar datos en una base de datos MongoDB. Algunas de las funcionalidades más comunes proporcionadas por PyMongo incluyen:

- **Conexión a la base de datos:** PyMongo permite establecer conexiones a una instancia de MongoDB, ya sea local o remota.
- **Inserción, actualización y eliminación de datos:** Puedes insertar nuevos documentos, actualizar registros existentes y eliminar datos de una base de datos MongoDB utilizando PyMongo.
- **Consultas (queries):** PyMongo te permite realizar consultas para recuperar datos de la base de datos, con opciones para filtrar, ordenar y limitar los resultados.
- **Índices:** Puedes definir y administrar índices en las colecciones de MongoDB para acelerar las consultas.
- **Aggregations (agregaciones):** PyMongo admite la realización de operaciones de agregación en datos almacenados en MongoDB.
- **Administración de colecciones y bases de datos:** Puedes crear, eliminar y administrar colecciones y bases de datos en MongoDB a través de PyMongo.

2.5.2 Visual Studio Code

Visual Studio Code, creado por Microsoft, es un entorno de desarrollo integrado (IDE) altamente adaptable. Es una elección destacada para el proyecto gracias a su facilidad de uso, su amplia

comunidad de extensiones y su capacidad de personalización (Microsoft, 2023). A continuación, se describe algunas características clave de Visual Studio Code:

- **Editor de texto avanzado:** VS Code proporciona un editor de texto altamente personalizable con funciones como resaltado de sintaxis, autocompletado inteligente, refactoring de código y muchas extensiones para mejorar la funcionalidad del editor (Microsoft, 2023).
- **Multiplataforma:** Puedes usar Visual Studio Code en Windows, macOS y Linux. Esto lo hace accesible para una amplia variedad de desarrolladores (Microsoft, 2023).
- **Extensiones:** Una de las características más destacadas de VS Code es su sistema de extensiones. Puedes ampliar las capacidades del IDE instalando extensiones que van desde lenguajes de programación adicionales hasta herramientas de desarrollo, integración con sistemas de control de versiones, depuración y mucho más. Hay una gran cantidad de extensiones disponibles en el mercado de extensiones de Visual Studio Code (Microsoft, 2023).
- **Integración con Git:** VS Code incluye integración nativa con el sistema de control de versiones Git, lo que facilita la gestión y el seguimiento de cambios en proyectos de desarrollo de software (Microsoft, 2023).
- **Depuración:** Puedes depurar tus aplicaciones directamente desde Visual Studio Code. Admite depuración para varios lenguajes de programación, incluyendo Python, JavaScript, C#, y muchos otros (Microsoft, 2023).
- **Terminal integrada:** VS Code incluye una terminal integrada que te permite ejecutar comandos y scripts directamente desde el IDE, lo que es especialmente útil para tareas de desarrollo y administración del sistema (Microsoft, 2023).

- **Integración en la nube:** Puedes integrar servicios en la nube, como Azure, directamente en Visual Studio Code para simplificar el desarrollo en la nube y la implementación de aplicaciones (Microsoft, 2023).
- **Comunidad activa:** Visual Studio Code cuenta con una comunidad activa de desarrolladores que contribuyen con extensiones y mejoras constantemente, lo que lo convierte en un IDE en constante evolución (Microsoft, 2023).

2.6 Técnicas de procesamiento de datos

El procesamiento de datos con Python se realiza comúnmente utilizando librerías y técnicas específicas para tareas como limpieza, manipulación, análisis y visualización de datos. A continuación, se presenta las librerías que utilizaremos en el procesamiento de datos con Python:

NumPy: Es una biblioteca esencial para el procesamiento numérico en Python, ofreciendo estructuras de datos eficientes para la manipulación de arreglos multidimensionales. Estas características son fundamentales en el análisis de datos y en cálculos matemáticos (Vasilev, 2019).

pandas: es una librería ampliamente utilizada para la manipulación y análisis de datos estructurados en forma de tablas. Ofrece DataFrames y Series que facilitan la carga, filtrado, agregación y transformación de datos.

Matplotlib y Seaborn: Estas librerías se utilizan para la visualización de datos. Matplotlib es una librería de trazado altamente personalizable, mientras que Seaborn proporciona una interfaz más simple y atractiva para crear gráficos estadísticos.

scikit-learn: es una librería de aprendizaje automático que ssce utiliza para tareas de modelado predictivo, clasificación, regresión, agrupación y preprocesamiento de datos.

SciPy: es una extensión de NumPy que proporciona funciones adicionales para optimización, estadísticas, procesamiento de señales y más.

NLTK (Natural Language Toolkit): es una librería para procesamiento de lenguaje natural que se utiliza en tareas como tokenización, análisis de sentimientos, etiquetado de entidades y análisis de texto.

SpaCy: es otra librería para procesamiento de lenguaje natural que se destaca por su rapidez y eficiencia en el análisis de texto en varios idiomas.

TensorFlow y PyTorch: Estas son dos librerías líderes en el campo del aprendizaje profundo (deep learning) y se utilizan para crear y entrenar modelos de redes neuronales artificiales

OpenCV (Open Source Computer Vision Library): es una librería para aplicaciones de visión por computadora. Es utilizada para procesar imágenes y videos, y para llevar a cabo tareas como detección de objetos, seguimiento y reconocimiento de patrones.

En Python, puedes cargar datos desde diversas fuentes. Las fuentes de datos más comunes incluyen:

Archivos Locales:

- **Archivos de texto**, como CSV, TXT o JSON.
- **Archivos de hojas de cálculo**, como XLSX (Excel).
- **Archivos de bases de datos**, como SQLite o MySQL.
- **Archivos en formato HDF5**.

Archivos Remotos: Puedes cargar datos directamente desde recursos en línea utilizando bibliotecas como requests para hacer solicitudes HTTP y luego procesar los datos.

Bases de Datos: Python tiene bibliotecas que permiten conectarse y consultar bases de datos, como MySQL, PostgreSQL, SQLite, MongoDB, y más. Ejemplos de estas librerías son SQLAlchemy, pymysql, psycopg2, sqlite3, y pymongo.

APIs Web: Puedes obtener datos de servicios web utilizando solicitudes HTTP y procesar la respuesta en formato JSON o XML.

Web Scraping: Puedes utilizar bibliotecas como BeautifulSoup y Scrapy para extraer datos de sitios web. Sin embargo, debes tener en cuenta las políticas de uso ético y legal al realizar web scraping.

Sensores y Dispositivos: Python se utiliza en la adquisición de datos a través de sensores y dispositivos en aplicaciones de IoT (Internet de las cosas). Puedes utilizar bibliotecas específicas para interactuar con sensores y dispositivos.

Streaming de Datos: En aplicaciones en tiempo real, como redes sociales o transmisiones en vivo, puedes utilizar bibliotecas y servicios que te permiten capturar y procesar datos en tiempo real, como tweepy para Twitter o Apache Kafka.

Archivos en la Nube: Algunos servicios de almacenamiento en la nube, como Amazon S3 o Google Cloud Storage, ofrecen APIs para acceder y cargar datos almacenados en la nube.

2.6.1 Pre procesamiento de datos

El preprocesamiento de datos es una parte fundamental en la preparación de datos para análisis, modelado de machine learning y otras tareas de procesamiento de datos. Python ofrece varias librerías y técnicas para llevar a cabo el preprocesamiento de datos. Aquí hay una guía general de las etapas comunes de preprocesamiento de datos en Python:

Carga de datos: Utiliza librerías como pandas para cargar tus datos desde diferentes fuentes, como archivos CSV, Excel, bases de datos, API web, etc.

Exploración de datos iniciales: Utiliza funciones de pandas para examinar los primeros registros, resúmenes estadísticos y detectar posibles problemas en los datos.

Limpieza y calidad de datos: Identifica y maneja valores faltantes en los datos, elimina duplicados, corrige errores en los datos, como errores tipográficos.

BUENAS PRÁCTICAS EN EL USO DE LA NORMA ISO/IEC 25012

La norma ISO/IEC 25012, conocida además como la norma de calidad de datos de software, nos proporciona directrices para evaluar y gestionar la calidad de los datos en sistemas de software (ISO-ORG, 2023). En el caso de este proyecto, las buenas prácticas que se aplicaran en relación con esta norma son:

Integridad de los datos: Se implementarán métodos de validación y restricción para evitar la entrada de datos incorrectos o incoherentes en la base de datos, garantizando de esta manera tener información real y que los datos sean coherentes en términos de formatos y valores (ISO-ORG, 2023).

Consistencia de los datos: Se mantendrá una estructura de datos coherente de la información de los médicos, para lo cual se usará convenciones de nomenclatura consistentes, campos estandarizados y formatos uniformes (ISO-ORG, 2023).

Completitud de los datos: Se asegurará que se tenga registrada en la base de datos la información relevante sobre los médicos. Se incluye detalles como nombres y apellidos completos, especialidades, información de contacto y el plan del laboratorio que tienen asignados (ISO-ORG, 2023).

Confidencialidad y seguridad de los datos: Debido que se trata de información personal de los médicos, se implementara medidas de seguridad con el fin de proteger la privacidad de los médicos y cumplir con las regulaciones de la ley de protección de datos (ISO-ORG, 2023).

Documentación detallada: Se documentará el diseño de la base de datos, con sus respectivos modelos E/R tanto físico como lógico, su correspondiente diccionario de datos, asi como las reglas de validación, los procesos de limpieza y carga de datos (ETL). Esto facilitará la comprensión y el mantenimiento de la base de datos a lo largo del tiempo (ISO-ORG, 2023).

Transformación de datos: Convierte datos categóricos en numéricos usando codificación one-hot (por ejemplo, con pandas o scikit-learn).

Escala características numéricas para que tengan una misma magnitud.

Realiza la ingeniería de características, creando nuevas variables o transformando las existentes.

Selección de características: Elimina características irrelevantes o redundantes.

Utiliza técnicas de selección de características, como la prueba estadística de chi-cuadrado o la importancia de características de modelos de machine learning.

Documentación y registro: Documenta los pasos de pre procesamiento y guarda información sobre cómo se manipularon los datos, lo que es importante para la reproducibilidad de los resultados.

2.7 Metodología ágil

La metodología ágil, también reconocida como enfoque ágil, constituye un conjunto de directrices y prácticas aplicadas en la gestión de proyectos y el desarrollo de software. Su propósito es poner de relieve la colaboración, la adaptabilidad y la entrega continua de productos de alta calidad. En contraste con las metodologías convencionales de gestión de proyectos, como el modelo en cascada, que se fundamentan en una planificación e implementación rígida y secuencial, el enfoque ágil se distingue por su flexibilidad y capacidad para adaptarse a cambios en los requisitos y prioridades a lo largo del ciclo de desarrollo.

Algunos de los principios clave de la metodología ágil, tal como se definen en el Manifiesto Ágil, incluyen:

1. Actores e interacciones sobre procesos y herramientas.
2. Software funcionando sobre documentos extensos.
3. Colaboración con clientes sobre negociaciones de contratos.
4. Respuesta a cambios sobre seguir un plan.

Existen varios marcos y metodologías ágiles populares, como Scrum, Kanban, Extreme Programming (XP) y Lean, que ofrecen enfoques específicos para implementar estos principios. Estas metodologías promueven la entrega incremental de software, la retroalimentación constante de los clientes, la autoorganización de equipos y la mejora continuala int.

2.7.1 Tipos de metodologías

Existen varias metodologías ágiles o enfoques que las organizaciones pueden adoptar para gestionar proyectos y desarrollar productos de manera más flexible y colaborativa. A continuación, se presentan algunos de los tipos de metodologías ágiles más populares

Scrum: Es uno de los métodos ágiles más ampliamente aceptados y se fundamenta en períodos de trabajo conocidos como "sprints", generalmente con una duración de 2 a 4 semanas. Durante cada sprint, se trabaja en el desarrollo de un conjunto específico de funcionalidades. Se focaliza en roles claramente definidos (Scrum Master, Product Owner y el Equipo de Desarrollo), reuniones periódicas y la entrega progresiva de incrementos funcionales.

Kanban: se centra en la gestión visual del trabajo. Las tareas se representan en tarjetas y se mueven a través de columnas en un tablero Kanban a medida que avanzan. No hay sprints o plazos fijos; en su lugar, Kanban se enfoca en la gestión del flujo de trabajo y la optimización continua.

Extreme Programming (XP): se enfoca en la mejora continua de la calidad del software y la satisfacción del cliente. Incorpora prácticas como la programación en parejas, pruebas unitarias, integración continua y lanzamientos frecuentes. XP pone un fuerte énfasis en la comunicación y la retroalimentación constante.

Lean Software Development: Creado en base a los principios del sistema de producción Lean, este enfoque se centra en eliminar lo no relevante, mejorar la eficiencia y entregar valor de manera más rápida. Se promueve la entrega de funcionalidades mínimas viables y la reducción de tiempos de espera.

Crystal: es una familia de metodologías ágiles desarrolladas por Alistair Cockburn. Ofrece una variedad de enfoques, desde Crystal Clear (para equipos pequeños y proyectos simples) hasta Crystal Orange (para proyectos más grandes y complejos). Se adaptan a las necesidades y características específicas del proyecto.

Dynamic Systems Development Method (DSDM): es una metodología ágil que pone un énfasis particular en la colaboración entre equipos de desarrollo y partes interesadas. Proporciona un marco de trabajo para el desarrollo rápido de sistemas y proyectos de TI.

Feature Driven Development (FDD): se centra en la descomposición de un sistema en pequeñas características o funcionalidades, que luego se desarrollan de manera incremental. Es especialmente útil en proyectos de desarrollo de software complejos.

Adaptive Project Framework (APF): es una metodología ágil que se centra en la adaptación continua a medida que los proyectos evolucionan. Se basa en la premisa de que cada proyecto es único y requiere un enfoque a medida.

Scaled Agile Framework (SAFe): es una metodología ágil diseñada para grandes organizaciones que desean escalar los principios ágiles a nivel empresarial. Proporciona un

conjunto de roles, prácticas y herramientas para la implementación ágil en organizaciones grandes y complejas.

2.7.1.1 Metodología Scrum

Un proyecto Scrum se puede entender como un trabajo en conjunto para desarrollar un producto nuevo, servicio u otros, dependiendo de lo establecido en la declaración de la visión del Proyecto. Los proyectos pueden verse afectados por distintos factores como: limitaciones temporales, económicas, organizacionales, de alcance, calidad, recursos u otras que hacen que sea complicada la planeación, ejecución, y administración del proyecto para finalmente tener éxito. Sin embargo, la implementación de un proyecto con éxito proporciona ventajas comerciales significativos a una organización. Por consiguiente, es primordial que las organizaciones elijan y practiquen un enfoque apropiado de manejo de proyectos.

Scrum es una de metodologías ágiles más populares. Es un marco adaptativo, iterativo, rápido, flexible y efectivo ideado para brindar un valor significativo a un proyecto. Scrum asegura la transparencia en la comunicación, creando un entorno de responsabilidad colectiva y progreso continuo.

Scrum se configura de forma que asiste en la elaboración de productos y servicios cubriendo industrias y proyectos, sin considerar su complejidad. Una de las principales ventajas de Scrum se basa en la utilización de equipos multifuncionales que se autoorganizan, permitiendo de esta manera organizar las tareas para tener periodos de trabajo cortos denominados *Sprints*. La Figura 7 da una visión general del flujo de un proyecto Scrum.

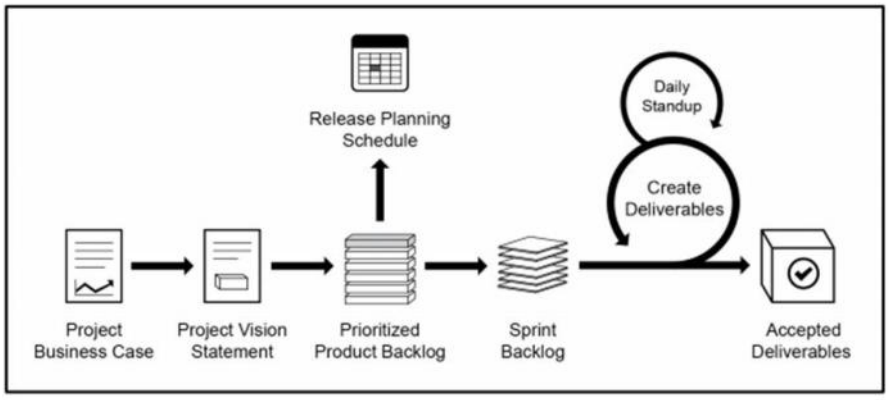
El periodo (iteración o ciclo) Scrum comienza con una reunión donde se involucran a todas las partes interesadas, durante la cual se define la visión del proyecto. Posterior a esto, el propietario del producto con el Scrum Master desarrolla el *Product Backlog* que es una lista de

requisitos ordenada por prioridad en forma de Historias de Usuarios. Cada *sprint* inicia con una reunión de planificación de *sprint* donde se toman las historias de usuarios con prioridad alta para su inclusión en el ciclo.

Un *sprint* implica que el equipo Scrum se enfoque en la creación de entregables o incrementos de producto potencialmente entregables. Durante el *sprint*, se realizan reuniones diarias cortas y totalmente enfocadas donde los miembros del equipo discuten el avance diario.

Finalizando el *sprint*, se realiza la reunión de revisión, donde el equipo Scrum se informa sobre el progreso, para luego realizar la actualización del *Product Backlog* (Carrión & Gómez, 2020).

Figura 7
Flujo Scrum para un sprint



Nota: Obtenido de https://upload.wikimedia.org/wikipedia/commons/thumb/e/e5/Scrumm.PNG/220px-Scrumm.PNG

El propietario del producto y las partes interesadas aceptan los entregables solo si cumplen con los criterios de aceptación predefinidos. (Gallego, 2019)

En la tabla 1 se enumeran los principales procesos Scrum que se pueden aplicar a cualquier tipo de proyecto, los mismos que se encuentran agrupados en 4 fases para una mayor comprensión (Carrión & Gómez, 2020).

Tabla 1
Fases Scrum

Fase	Proceso SCRUM fundamental
Inicial	Crear la visión del proyecto, que servirá de inspiración y proveerá una guía clara durante todo el proyecto Identificar al *Scrum Master* (persona que lidera y asesora a los equipos) y a las partes interesadas, mediante un criterio específico. Formar el equipo Scrum. El propietario del producto y el Scrum Master se encargan de la selección de los miembros del equipo. Crear las historias de usuario, diseñadas por el propietario del producto con la ayuda del equipo Scrum para asegurarse que los requerimientos del cliente estén claramente plasmados y

	completamente comprendidos por las partes interesadas; posteriormente estas historias se incorporan al *Product Backlog*.
Planeación y Estimación	Identificar Tareas. En este proceso se transforma la historia de usuario en una lista de tareas específicas, y se evalúa el esfuerzo estimado que éstas conllevarán. El equipo Scrum revisa estas historias con el propósito de generar un calendario de planificación de entrega, que es básicamente un calendario de implementación que puede ser compartido con los interesados. Crear el *Product Backlog*, que contiene los requerimientos del proyecto definidos y priorizados. Crear *sprint Backlog*. En este proceso el equipo Scrum crea el *sprint backlog* que contiene todas las tareas a ser completadas en un *sprint* como parte de la reunión de planificación
Implementación	Crear los entregables. En este proceso el equipo Scrum trabaja en las tareas que se cargaran en el *sprint backlog* para definir los entregables del *Sprint*. Se utiliza usualmente un *Scrum Board* (tablero de organización) para llevar un seguimiento del trabajo y actividades realizadas, así como también los inconvenientes que se hayan presentado. Realizar la reunión de revisión. En esta reunión el equipo Scrum comparte los progresos realizados y los inconvenientes presentados que deben ser solucionados. Posteriormente se realiza la actualización del *Product Backlog* con cualquier cambio o modificación debatido en la reunión.
Revisión y Retrospectiva	Demostrar y Validar el *sprint*. En este proceso el equipo Scrum presenta el entregable del *sprint* al propietario del producto y a las partes interesadas, con el fin de asegurar la aprobación por parte de estos.

Nota: *Datos tomados de la tesis Desarrollo De Un Prototipo Para Control De Acceso A Aulas Y Registro De Asistencia Del Personal Docente De La Fiee 2020 (p.18), por Gómez Doménica,2020*

2.8 Protección de datos personales

2.8.1 Aspectos regulatorios según normativa internacional

La protección de nuestros datos personales es fundamental y está garantizada a través de leyes específicas. La UE fue una de las primeras en establecer normativas en este campo en los 90 y ha seguido mejorándolas. Otros países han seguido su ejemplo, como Brasil y Ecuador.
Sin embargo, muchos otros aún no cuentan con leyes efectivas y a pesar de los escándalos relacionados con la privacidad, siguen sin proteger adecuadamente los datos personales. Algunos países han aprobado leyes, pero no las respetan, mientras que, en otros, a pesar de tener leyes sólidas, la implementación resulta ser más complicada de lo previsto.

Unión Europea

La Unión Europea se destaca como un referente global en la protección de datos personales, y su influencia está creciendo a medida que implementa su reglamento, lo que anticipa una mayor consideración por parte de empresas y autoridades en el futuro (Parlamento Europeo, 2016). Los elementos clave de este reglamento incluyen la precisa identificación de la finalidad del tratamiento, el requerimiento de un consentimiento libre, específico, informado e inequívoco, la obtención de información más extensa que la establecida en la Ley Orgánica de Protección de Datos (LOPD), la creación de un registro de actividades, la notificación inmediata de cualquier brecha de seguridad o violación, la implementación de una metodología para la evaluación de riesgos, la incorporación de la figura del Data Protection Officer (DPO), la introducción de certificaciones y sellos de cumplimiento, la realización de transferencias internacionales con garantías y el fortalecimiento del régimen sancionador.

El Reglamento General de Protección de Datos Personales, RGPD, de la Unión Europea es una de las iniciativas más avanzadas. Ha motivado a muchos países a adoptar leyes de protección de datos personales o adaptar las existentes. A su vez, ha impuesto grandes multas sobre las empresas que infringieron los derechos de las personas. En 2021, Amazon recibió una multa de 746 millones de euros por incumplir el RGPD.

Sin embargo, tal como lo demuestra nuestro informe de 2021 sobre la implementación del RGPD, la mayoría de los gobiernos han sido poco expeditivos en la aplicación de la ley, muchas empresas aún lo ignoran, y las personas todavía no notan cambios en cómo se protegen sus derechos.

Otro caso muy reciente, el pasado 5 de enero de 2023, salió la noticia de que Bruselas había colocado una multa considerada histórica a la compañía Meta que anteriormente se denominaba Facebook por anuncios publicitarios que eran obligatorios para sus usuarios. La compañía deberá pagar 390 millones de euros por incumplimiento de la normativa comunitaria, además Meta deberá cambiar en Europa su modelo publicitario y, por ende, su modelo de negocio.

Estados Unidos

La cuestión de la protección de datos en los Estados Unidos presenta una complejidad notable. La salvaguarda de datos personales exhibe notables variaciones entre países, algunos de los cuales cuentan con normativas más flexibles, especialmente tras la implementación del Reglamento General de Protección de Datos (RGPD). En contraste, en Estados Unidos, las normas y regulaciones para el manejo de datos difieren entre los estados, resultando en niveles distintos de seguridad y requisitos para las empresas según su ubicación operativa.

En California, se aprobó la Ley de Privacidad del Consumidor de California (CCPA) en 2020, inspirada en el RGPD de la Unión Europea, representando un paso significativo hacia una regulación más integral en el país. Asimismo, Estados Unidos ha adoptado un conjunto mínimo de estándares de privacidad en línea con las directrices establecidas por APEC, la Asociación de Asia y el Pacífico que incluye a 21 países como Japón, Canadá y México. Los estándares APEC se exploran con mayor detalle en secciones subsiguientes.

La CCPA de California impone a las empresas la responsabilidad de gestionar de manera segura los datos personales, requiriendo además que los usuarios estén informados acerca del uso de dichos datos y notificándoles en caso de cualquier compromiso de la seguridad de los mismos. Además, existen normativas específicas para el tratamiento de información médica, como lo establece la legislación HIPAA, que obliga a las empresas que manipulan información médica protegida a seguir medidas de seguridad físicas en todas sus redes y procesos.

México

En México, la gestión de la información personal está sujeta a regulaciones establecidas por la Ley Federal de Protección de Datos Personales en Posesión de los Particulares (LFPDPPP) desde julio de 2010, la cual rige el tratamiento de datos personales por parte de entidades del sector privado. Asimismo, desde 2017, se implementa la Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados para abordar el tratamiento de datos en el sector público.

En este contexto, México dispone de leyes específicas para la protección de datos y cuenta con una entidad reconocida por la International Conference of Data Protection and Privacy Commissioners (ICDPPC). Ambas leyes regulan el manejo de datos personales y requieren el consentimiento del titular. También establecen normativas respecto al derecho de cancelación de datos y pueden imponer sanciones que oscilan entre advertencias y multas de cuantía variable, dependiendo del salario mínimo en el Distrito Federal.

Argentina

En Argentina, la Ley de Protección de Datos Personales (PDPA) fue promulgada en el año 2000, convirtiendo al país en el primero de América Latina en poder realizar transferencias de datos con naciones de la Unión Europea sin la necesidad de herramientas adicionales. En 2016, la Agencia de Acceso a la Información Pública aprobó un nuevo reglamento para la transferencia de datos personales más allá de las fronteras, el cual demanda el consentimiento explícito para el tratamiento de datos, reconoce los derechos de supresión, rectificación y actualización de la información, y establece sanciones, ya sean administrativas o penales, como resultado de las modificaciones introducidas al Código Penal.

Brasil

La Ley General de Protección de Datos (LGPD) de Brasil, implementada en 2020, establece la necesidad de obtener consentimiento para el procesamiento de datos personales y especifica situaciones en las que no es obligatorio. Garantiza el derecho a la eliminación de datos y tiene la autoridad para aplicar sanciones que pueden llegar hasta el 2% de la facturación anual de la empresa o un máximo de 50 millones de reales.

Colombia

La Ley 1581 de 2012 constituye la base principal de la legislación en Colombia referente a la protección de datos personales. Colombia ha continuado perfeccionando sus disposiciones normativas y cuenta con una entidad independiente, situándola en una posición similar a la de México.

Algunos aspectos significativos de esta ley incluyen la imposición de requisitos para el manejo de datos personales, que abarcan la necesidad de obtener la autorización del titular y el reconocimiento de derechos para aquellos que proporcionan sus datos. En caso de incumplimiento, las sanciones pueden llegar hasta los 2000 salarios mínimos legales, lo que equivale a aproximadamente 1850 millones de pesos colombianos.

Chile

La protección de datos personales en Chile se rige por la Ley de Protección de la Vida Privada (LPVP) desde 1999. A lo largo del tiempo, se han realizado modificaciones y actualmente se están realizando esfuerzos para desarrollar una ley específica dedicada a la protección de datos. Algunos aspectos a considerar incluyen la obligatoriedad de obtener autorización por escrito para el tratamiento de datos, la redacción poco clara respecto a los derechos de acceso, rectificación, cancelación y oposición, la ausencia de un ente supervisor, y las sanciones que oscilan entre 1 y 50 unidades tributarias mensuales, siendo las infracciones relativamente leves. Actualmente, se están implementando mejoras en estas áreas.

UK (Reino Unido)

La Ley de Protección de Datos del Reino Unido (DPA) es la implementación del Reglamento General de Protección de Datos (RGPD) y entró en vigor el 23 de mayo de 2018. DPA incluye

datos personales, incluidos los datos de delincuentes, y permite la creación de perfiles de personas. Además, se extiende para incluir recopilación de inteligencia, inmigración y las autoridades. Las multas para los infractores reincidentes son estrictas y cuantiosas.

Por otro lado, el NHS, sistema nacional de salud, tiene reglas específicas para organizaciones que trabajan con datos de salud.

APEC (Foro de Cooperación Económica Asia-Pacífico)

APEC es una organización de 21 países de Asia Pacífico, incluyendo Estados Unidos, Japón, Corea del Sur, Canadá y México. Las Reglas de Privacidad Transfronterizas (CBPR) establecidas por APEC proporcionan una base para las leyes de privacidad dentro de cada uno de los países miembros, y por el momento lo han adoptado EE.U.U., México, Japón y Canadá. CBPR se aplica a cualquier organización pública o privada que maneje datos personales, pero solo a los controladores, no a los procesadores.

CBPR está destinado a proporcionar un nivel mínimo de protección y pende de los Estados miembros para construir sobre ese marco con reglas para sus mercados específicos. Estados Unidos ha aceptado el marco CBPR y puede dar una pista sobre qué tipos de legislación sobre privacidad de datos pueden provenir del Congreso de los Estados Unidos.

Corea del Sur

La Ley de Protección de Información Personal (PIPA) entró en vigor el 30 de septiembre de 2011 y se une a un conjunto de leyes de seguridad de datos que son quizás las más estrictas del mundo. Con esta ley se protege cualquier dato que pueda identificar a una persona, incluidas las imágenes. También hace una distinción para los datos personales "sensibles", como la religión o la orientación sexual, que podrían utilizarse para infringir los derechos personales.

Además, Corea del Sur tiene leyes de privacidad de datos específicas para sectores como TI, información crediticia y finanzas.

La definición de información personal incluye cualquier dato que pueda identificar a una persona, incluyendo datos parciales que podrían combinarse con otros para identificar a una persona en particular.

Por ejemplo, si los datos anonimizados pudieran combinarse con otras fuentes de datos para identificar a una persona, los datos anonimizados tendrían que manejarse con la misma precaución que cualquier otro dato personal.

China

La Ley de Seguridad Cibernética (CSL) de China entró en vigor en junio de 2017 y regula todos los datos personales de ciudadanos chinos.

No está permitido el almacenamiento de datos personales en el extranjero, excepto con una justificación documentada y una evaluación de seguridad. La CSL se aplica a manejadores de datos y operadores de telecomunicaciones, radio y televisión. Las autoridades chinas deben ser informadas si los datos indican actividades ilegales, por lo que, paradójicamente, pueden ser examinados por las propias autoridades.

2.8.2 Aspectos regulatorios según normativa local

Después de un esfuerzo de varios años liderado por la Dirección Nacional de Registro de Datos Públicos (DINARDAP) y con la participación de diversas entidades, desde mayo de 2021 Ecuador cuenta con su primera Ley Orgánica de Protección de Datos Personales (LOPDP). Esta normativa promueve el desarrollo de la innovación y el uso de tecnología, protegiendo en su centro el tratamiento de los datos personales (Dirección_Nacional_de_Registros_Públicos, 2021).

La Constitución del Ecuador reconoce y garantiza en el artículo 66 epígrafe 19 a las personas:

> *"El derecho a la protección de datos de carácter personal, que incluye el acceso y la decisión sobre información y datos de este carácter, así como su correspondiente protección. La recolección, archivo, procesamiento, distribución o difusión de estos datos o información requerirán la autorización del titular o el mandato de la ley."* (Dirección_Nacional_de_Registros_Públicos, 2021)

Así es, la LOPDP es una ley efectiva para la protección de los datos personales en Ecuador, desde mayo de 2021, cuando se adoptó la ley. La ley fue diseñada después de considerar los aportes de distintos actores, incluida la sociedad civil, y es considerada como una ley moderna. La aplicación de esta ley es crucial para garantizar la protección de los datos personales en el país y puede ser utilizada como modelo para la región.

El propósito principal de la ley es mejorar la seguridad de la información relacionada con los datos personales de las organizaciones, proteger las bases de datos de las empresas que contienen información sobre sus clientes con datos que permiten identificarlos y establecer la confidencialidad de los datos personales para evitar su uso con otros fines.

La normativa se aplica a todas las empresas públicas y privadas que traten datos personales en Ecuador, ya sea a través de la oferta de bienes o servicios, contratos o regulaciones internacionales. La Ley cuenta con 13 Principios: Juridicidad, Lealtad, Transparencia, Finalidad, Pertenencia, Proporcionalidad, Confidencialidad, Calidad, Conservación, Seguridad, Responsabilidad, Aplicación favorable al titular e Independencia del control.

De igual modo presenta unos derechos nuevos como:

- Derecho a la información
- Derecho de acceso
- Derecho de rectificación y actuación
- Derecho de eliminación

- Derecho de oposición
- Derecho a la portabilidad
- Derecho a la suspensión del tratamiento
- Derecho a la suspensión del tratamiento (limitación de uso)
- Derecho a no ser objeto de una decisión basada en valoraciones automatizadas
- Derecho de consulta en los registros de protección de datos
- Derecho a la educación digital

La Ley asigna responsabilidades para la aplicación de la normativa de protección de datos, estableciendo un responsable del tratamiento de datos, un delegado de Protección de Datos (DPD), un responsable y un encargado de tratamiento de datos. La persona designada como DPO es la encargada de informar a la entidad sobre sus obligaciones legales y de supervisar el cumplimiento de la normativa (Dirección_Nacional_de_Registros_Públicos, 2021).

La entidad responsable debe implementar políticas de protección de datos, evaluar el nivel de seguridad previo al tratamiento de datos personales, suscribir contratos de confidencialidad y designar al DPD.

Las organizaciones deben implementar medidas de seguridad para proteger adecuadamente los datos personales, incluyendo anonimización, integridad, confidencialidad y resiliencia técnica, física, administrativa y jurídica. Además, deben llevar a cabo una verificación continua y permanente de la eficiencia y efectividad de las medidas de seguridad (Dirección_Nacional_de_Registros_Públicos, 2021).

En caso de una vulneración de datos, la entidad responsable debe notificar al titular y a la autoridad de protección de datos en un plazo de 3 y 5 días, respectivamente, después de tener

conocimiento de la vulneración y si conlleva un riesgo a los derechos fundamentales y libertades individuales del titular (Dirección_Nacional_de_Registros_Públicos, 2021).

Además, la Ley de Protección de Datos Personales incluye sanciones y multas para los servidores públicos, responsables y encargados de manejar los datos que no cumplan con sus regulaciones. Estas sanciones van desde 1 hasta 20 salarios unificados en caso de multas graves para servidores públicos, y para responsables o encargados van desde 0.7% hasta 1% calculado en base al volumen de negocios del ejercicio económico anterior (Dirección_Nacional_de_Registros_Públicos, 2021).

2.9 Análisis Pestel

El análisis PESTEL es una herramienta de análisis estratégico utilizada en la gestión empresarial para evaluar el entorno externo en el que opera una organización (Fred R, 2011). La sigla PESTEL se refiere a seis factores clave que se analizan para comprender el contexto en el que opera una empresa. Aquí están las definiciones de cada uno de los componentes del análisis PESTEL:

Político (Political): Este factor se refiere a la influencia de los aspectos políticos y gubernamentales en el entorno empresarial. Incluye la estabilidad política, las políticas gubernamentales, las regulaciones, la legislación y los impuestos. Un análisis político examina cómo las decisiones gubernamentales y la estabilidad política pueden afectar a una organización y su industria (Johnson Gerry, 2005).

Económico (Economic): El factor económico se enfoca en las condiciones económicas que pueden afectar a una empresa. Esto incluye indicadores como la inflación, el crecimiento económico, las tasas de interés, las tasas de cambio y el ciclo económico. Un análisis

económico evalúa cómo estas condiciones pueden influir en la demanda, los costos y la rentabilidad de una empresa (Johnson Gerry, 2005).

Social (Social): El análisis social se centra en los aspectos demográficos y socioculturales de la sociedad que pueden afectar a una organización. Esto incluye la demografía de la población, las tendencias culturales, los valores, las preferencias del consumidor y las dinámicas sociales. Un análisis social ayuda a entender las necesidades y deseos cambiantes de los consumidores (Johnson Gerry, 2005).

Tecnológico (Technological): Este factor considera el impacto de la tecnología en la industria y en la empresa en particular. Incluye avances tecnológicos, innovaciones, tasas de adopción de tecnología y desarrollos en investigación y desarrollo. Un análisis tecnológico examina cómo la tecnología puede influir en la eficiencia, la competencia y las oportunidades de crecimiento (Johnson Gerry, 2005).

Medioambiental (Environmental): Este componente se refiere a las preocupaciones y regulaciones relacionadas con el medio ambiente y la sostenibilidad. Incluye cuestiones como el cambio climático, la gestión de residuos, la sostenibilidad y las normativas ambientales. Un análisis medioambiental evalúa cómo las prácticas empresariales pueden impactar en el entorno y cómo las regulaciones medioambientales pueden afectar la empresa (Johnson Gerry, 2005).

Legal (Legal): El factor legal considera las leyes y regulaciones que afectan a una organización. Esto incluye regulaciones comerciales, laborales, de salud y seguridad, de propiedad intelectual y otras normativas legales. Un análisis legal examina cómo las regulaciones y las cuestiones legales pueden influir en las operaciones y la estrategia de la empresa (Johnson Gerry, 2005).

CAPITULO III METODOLOGÍA

3.1 Fuentes de información

Las fuentes de información que se utilizarán en el proyecto serán:

3.1.1 Fuentes internas

Como fuentes de información interna tenemos la información en archivos Excel, debido a que la organización no dispone de una herramienta informática (ERP, CRM) en la cual los visitadores puedan registrar la información, adicional existen ciertos controles y estrategias de venta que impiden la implementación de una herramienta transaccional. Se dispone de archivos Excel, los cuales son manejados por los supervisores, ellos a su vez consolidan la información recolectada por los visitadores relacionada a la información personal de los médicos, posterior a ello asignan a los visitadores o representantes y planes con los cuales se realizarán las respectivas visitas.

3.1.2 Fuentes externas

Una vez implementado el proceso, se pueden considerar fuentes externas de información con el objetivo de garantizar la calidad y veracidad de los datos que se insertan en la base. Entre las posibles fuentes se encuentran bases de datos de acceso público, como Senescyt, el Ministerio de Salud y el Registro Civil.

3.2 Estructura base de datos relacional

Las tablas que intervienen para este proyecto son:

Tabla 2
Listado de tablas de la base de datos SQL

Tablas	Listado de campos	Identificar cantidad de registros	Tipo de campos
Especialidades	* Código de la especialidad * Nombre de la especialidad	43	*Numérico * Texto
Visitadores	* Código del visitador * Apellido Paterno * Apellido Materno * Nombres * Cédula de identidad * Código del supervisor	220	* Numérico * Texto * Texto * Texto * Numérico * Numérico
Supervisor	* Código del supervisor * Apellido Paterno * Apellido Materno * Nombres * Cédula de identidad	20	* Numérico * Texto * Texto * Texto * Texto
Planes	* Código del plan * Nombre del plan	26	* Numérico * Texto
Médico	* Código del médico * Apellido Paterno * Apellido Materno * Nombres * Número de identificación * Fecha de nacimiento * Correo * Celular * Dirección * Especialidad del médico	8000	* Numérico * Texto * Texto * Texto * Texto * Texto * Texto * Texto * Texto * Texto
Plan Visitador	* Código plan visitador * Código plan * Código visitador	330	* Numérico * Numérico * Numérico
Plan Visitador Médico	* Código plan visitador médico * Código plan visitador * Código médico	12500	* Numérico * Numérico * Numérico

3.2.1 Estructura base de médicos

Almacena la información referente a los datos personales de los médicos.

Tabla 3
Estructura Base de médicos

Código	Descripción	Primary Key	Foreign Key
id_medico	Identificador único del medico	X	
apellido_paterno	Apellido paterno del medico		
apellido_materno	Apellido materno del medico		
nombres	Nombres del medico		
numero_identificacion	Número de identificación del medico		
fecha_nacimiento	Fecha de nacimiento del medico		
email	Correo electrónico del medico		
celular	Numero celular del medico		
dirección	Dirección del medico		
id_especialidad	Clave foránea de la tabla que representa al id de la especialidad del medico		X

3.2.2 Estructura Base de Especialidades

Almacena la información de las especialidades que tiene cada médico

Tabla 4
Estructura Base de especialidades

Código	Descripción	Primary Key	Foreign Key
id_especialidad	Identificador único de la especialidad.	X	
nombre_especialidad	Nombre de la especialidad		

3.2.3 Estructura Base de Planes

Almacena la información acerca de los planes que va a tener cada médico.

Tabla 5
Estructura de la Base de Planes

Código	Descripción	Primary Key	Foreign Key
id_plan	Identificador único del tipo de identificación.	X	
nombre_plan	Nombre del plan		

3.2.4 Estructura Base de Visitadores

Almacena la información referente a los datos de los visitadores médicos.

Tabla 6
Estructura Base de visitadores

Código	Descripción	Primary Key	Foreign Key
id_visitador	Identificador único del visitador.	X	
apellido_paterno	Apellido paterno del visitador.		
apellido_materno	Apellido materno del visitador.		
nombres	Nombres del visitador		
id_tipo_identificacion	Clave foránea de la tabla que representa al id del tipo de identificación del visitador		X
numero_identificacion	Número de identificación del visitador		
id_supervisor	Clave foránea de la tabla que representa al id del supervisor		X

3.2.5 Estructura Base de Supervisores

Almacena la información referente a los datos de los supervisores de los visitadores.

Tabla 7
Estructura Tabla de Supervisor

Código	Descripción	Primary Key	Foreign Key
id_supervisor	Identificador único del supervisor.	X	
apellido_paterno	Apellido paterno del supervisor		
apellido_materno	Apellido materno del supervisor		
nombres	Nombres del supervisor		
id_tipo_identificacion	Clave foránea de la tabla que representa al id del tipo de identificación del supervisor		X
numero_identificacion	Número de identificación del supervisor		

3.2.6 Estructura Base de Planes-Visitadores

Tabla de cruce que almacena la información referente a los datos de los planes de que tienen asignados los visitadores.

Tabla 8
Estructura Base de Planes-Visitador

Código	Descripción	Primary Key	Foreign Key
id_plan_visitador	Identificador único de la tabla planes visitadores.	X	
id_plan	Clave foránea de la tabla que representa al id del plan		X
id_visitador	Clave foránea de la tabla que representa al id del visitador		X

3.2.7 Estructura Tabla Planes-Visitadores-Médicos

Tabla de cruce que almacena la información referente a los datos de los planes de los visitadores asignados a los médicos.

Tabla 9
Estructura Base de Planes-Visitador-Médicos

Código	Descripción	Primary Key	Foreign Key
id_plan_visitador_medico	Identificador único de la tabla planes visitadores medicos.	X	
id_plan_visitador	Clave foránea de la tabla que representa al id del plan visitador.		X
id_medico	Clave foránea de la tabla que representa al id del médico		X

3.3 Diccionario de datos

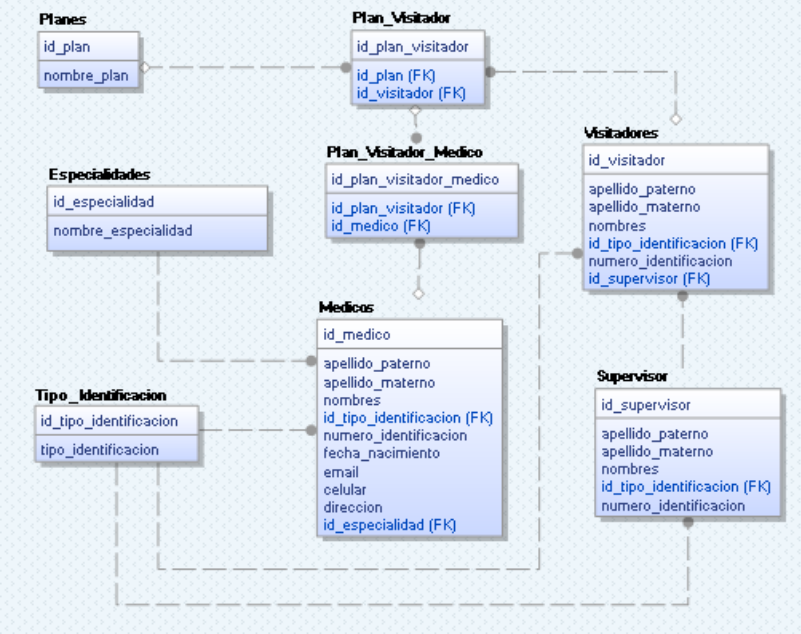
Tablas del Diagrama Médicos Laboratorio Farmacéutico

Tabla 10
Diccionario de datos

N°	Nombre	Descripción
1	Medicos	Tabla que almacena la información referente a los datos personales de los médicos.
2	Especialidades	Tabla que almacena la información de las especialidades que tiene cada medico
3	Tipo_Identificacion	Tabla que almacena la información de los tipos de identificación que puede tener
4	Planes	Tabla que almacena la información acerca de los planes que va a tener cada médico.
5	Visitadores	Tabla que almacena la información referente a los datos de los visitadores médicos.
6	Supervisores	Tabla que almacena la información referente a los datos de los supervisores de los visitadores.
7	Planes_Visitadores	Tabla de cruce que almacena la información referente a los datos de los planes asignados a los visitadores.
8	Planes_Visitadores_Medicos	Tabla de cruce que almacena la información referente a los datos de los planes de los visitadores asignados a los médicos.

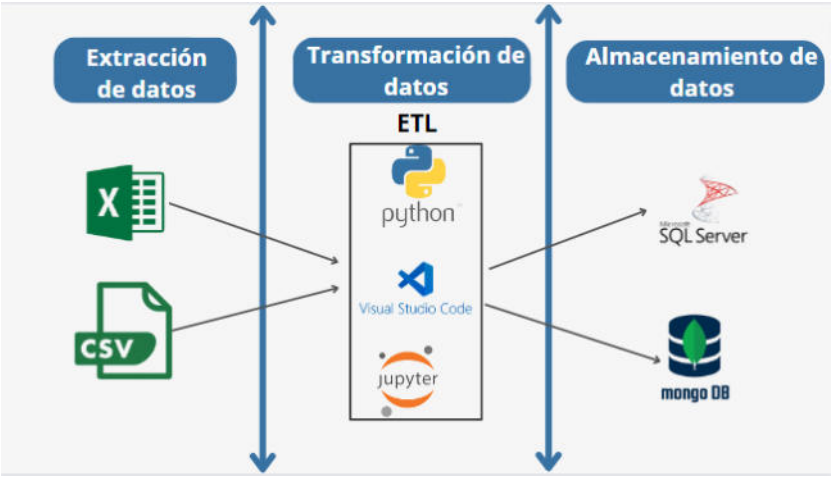
3.4 Diagrama entidad-relación

Tabla 11
Diagrama entidad-relación de las bases de datos



3.5 Arquitectura del proyecto

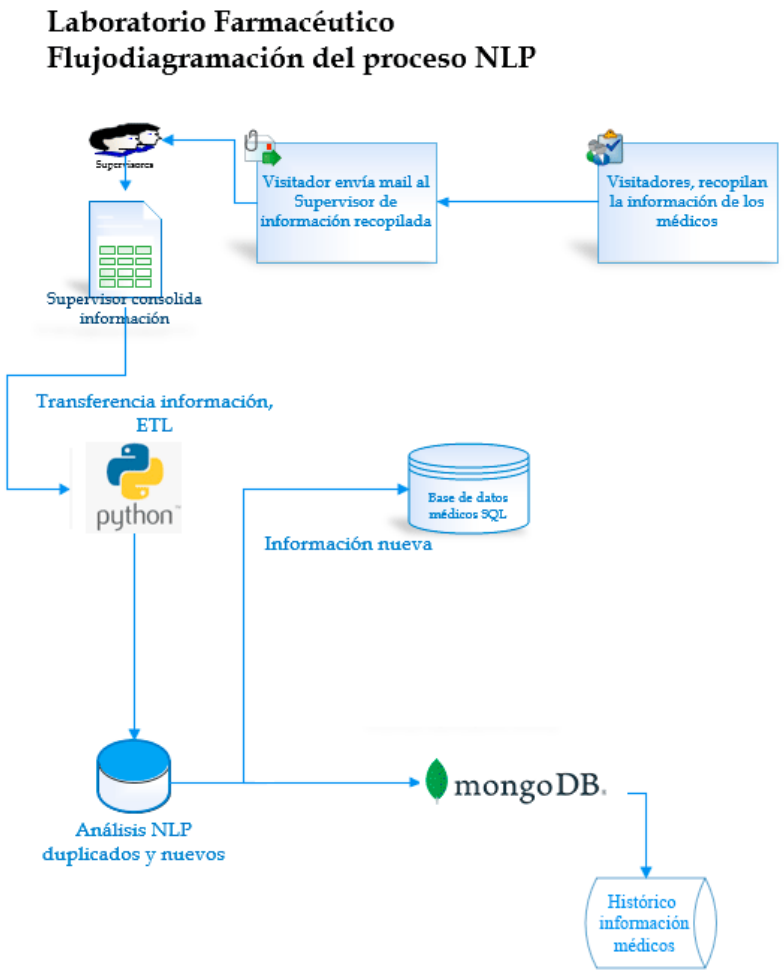
Tabla 12
Arquitectura del proyecto



3.6 Flujo del proceso del proyecto

Toda la información consolidada por los supervisores será cargada en la base de datos no relacional mediante un proceso ETL, posteriormente el modelo NLP leerá la información contenida en la tabla de médicos de la base NoSQL y realizará una comparación con la información que contiene la base de datos relacional, evitando así la duplicidad de la información e ingresos erróneos de nuevos médicos en el sistema.

Tabla 13:
Flujograma del proceso



- Visitadores recopilan la información referente a los médicos
- Visitadores envían la información recopilada a los supervisores
- Los supervisores consolidan la información de sus visitadores mediante el documento Excel programado que se incorpora como parte de este proyecto para control de calidad de información.
- Se realiza la subida de la información consolidada enviada por los supervisores a la base de datos no relacional mediante un proceso ETL.
- Se ejecuta el modelo NLP para contrastar la información de la base de datos NoSQL con la base de datos transaccional en cuanto a los nombres del médico.
- Inserción a la base de datos SQL de los médicos nuevos detectados mediante el proceso de NLP.
- Paso de información de los resultados NLP a las tablas de la base de datos NoSQL.
- Generación de reportes de respaldo del proceso ejecutado con médicos nuevos y duplicados obtenidos del modelo NLP.

3.7 Procesos de transformación de datos

Para el proceso de transformación de datos se utilizará un ETL, en donde la información enviada en Excel por parte de los supervisores será ingresada a Python, posterior a ello se correrá un proceso de limpieza y calidad de datos detallada en el siguiente punto y luego con la información de esta tabla se procederá con el modelo de NLP.

- Eliminación de espacios en blanco al inicio y al final de los campos nombres y apellidos de los médicos.
- Eliminación de caracteres especiales en todos los campos excepto en el correo y dirección.

- Al disponer un archivo Excel programado en visual basic se mejorará la calidad de los datos, debido a que existirán campos con características predefinidas reduciendo así errores de digitación.
- En el campo cédula de identidad se correrá un algoritmo verificador de identificaciones que permite detectar si una cédula es válida.

3.8 Modelos y algoritmos

Para el desarrollo del proyecto se considerará el uso de técnicas avanzadas de NLP, como modelos de lenguaje como BERT, así como también los modelos de FuzzyWuzzy es utilizado para medir la similitud entre dos cadenas. Es útil en tareas de coincidencia de registros, duplicación de datos, entre otros usos relacionados con el análisis de texto.

Algunos de los algoritmos más comunes incluidos en FuzzyWuzzy son:

- Ratio: Este algoritmo calcula la distancia de edición según el orden de ambas cadenas completas.
- Partial Ratio: Calcula la puntuación de similitud en función de la longitud de la subcadena común más larga, en lugar de la longitud de la cadena completa.
- Token Sort Ratio: Este algoritmo compara cadenas después de dividirlas en tokens y ordenar estos tokens alfabéticamente. Luego, calcula la relación entre los tokens comunes en ambas cadenas.
- Token Set Ratio: Similar a Token Sort Ratio, pero en lugar de ordenar los tokens alfabéticamente, los compara como conjuntos. Es útil cuando el orden de los elementos no es relevante.

Para todos los algoritmos de inteligencia artificial se establecerá un umbral de aceptación del 80% para evaluar el modelo. Cualquier par de registros que coincidan en un grado superior al

80% se considerarán duplicados. Por otro lado, aquellos que tengan una similitud inferior al 80% se clasificarán como registros correspondientes a nuevos médicos y se procederá a su inclusión en la base de datos.

3.9 Análisis PESTEL

3.9.1 Ámbito político

Impacto del nuevo gobierno. Las futuras elecciones a nivel nacional suscitan anticipaciones en el ámbito de la salud, las cuales podrían tener repercusiones significativas en la industria farmacéutica. En las propuestas de ambos candidatos, se destaca la priorización de abordar la problemática de la desnutrición materno-infantil, así como el aumento en la contratación de profesionales de la salud, como médicos, enfermeras y personal sanitario, especialmente en regiones rurales desatendidas, con el fin de fortalecer la atención primaria.

Política de igualdad de género y no discriminación: La empresa implementa políticas transversales de igualdad de género y no discriminación. Como resultado de estos esfuerzos, actualmente la mayor parte de sus colaboradores son mujeres, los cargos de jefaturas también son ocupados por mujeres. De esta forma, se sigue impulsando el crecimiento y bienestar de los colaboradores, promoviendo la inclusión y el empoderamiento femenino, con igualdad de oportunidades para todos, cumpliendo con el artículo 331 de la constitución ecuatoriana.

Regulación farmaceuticas: De acuerdo con el artículo 28 de la Ley Orgánica de Prevención Integral del Fenómeno Socioeconómico de las Drogas y de Regulación y Control de Uso de Sustancias Catalogadas Sujetas a Fiscalización, se otorgan facultades para supervisar y regular a la Autoridad Sanitaria Nacional en lo que respecta a la prescripción de medicamentos. El laboratorio farmacéutico dispone de dos categorías de productos: los llamados "Farma", que

incluyen todos los medicamentos recetados bajo prescripción médica y cuyos precios están sujetos a regulación por parte del Gobierno Nacional, y los denominados "Venta Libre".

3.9.2 Ámbito económico

Competencia justa: Respetamos el principio de libre competencia en los mercados y cumplimos con la Ley Orgánica de Regulación y Control del Poder de Mercado en sus artículos 35 y 336. Creemos que la competencia justa, en base a calidad, precio y servicio es beneficiosa para los consumidores. Promovemos un mercado libre, justo y leal. No participamos en acuerdos ni prácticas con nuestros competidores que impidan o limiten la libre competencia del mercado. Creemos en la igualdad de oportunidades y rechazamos acceder ilícitamente a información del mercado y de nuestros competidores.

3.9.3 Ámbito social

Brigadas Médicas: Consistente con su misión empresarial, la compañía impacta positivamente a numerosas personas cada mes mediante la realización de jornadas médicas gratuitas a nivel nacional. Durante estas brigadas, la población tiene la oportunidad de someterse a evaluaciones de hígado graso y densitometría ósea. Este esfuerzo, focalizado principalmente en las comunidades más desfavorecidas, refleja el compromiso social de la empresa.

Campañas Con Propósito: Con la finalidad de fomentar hábitos de vida saludables y reconocer los esfuerzos y sacrificios individuales, se llevan a cabo campañas con objetivos definidos, destinando beneficios especialmente a grupos vulnerables.

3.9.4 Ámbito tecnológico

Con el propósito de mejorar la eficiencia operativa y fortalecer la relación con los clientes, se establecieron colaboraciones estratégicas con empresas que aportan innovación tecnológica. La

herramienta resultante ofrece capacidades de personalización y segmentación de clientes, siendo esencial para analizar datos de usuarios, comprender sus necesidades y preferencias, y proporcionarles recomendaciones y ofertas adaptadas a sus requerimientos. Además, los clientes disfrutarán de beneficios como la facilidad de navegación, búsqueda de productos, realización de pedidos y seguimiento de su estado en la plataforma, así como promociones exclusivas y comunicación directa con el laboratorio. Con esta iniciativa, la empresa reitera su compromiso con la transformación digital, consolidándose como una entidad farmacéutica avanzada.

Avances en NLP: Mantenerse actualizado con los avances en tecnología NLP es esencial para garantizar las mejores prácticas y las herramientas más avanzadas, ya que esto puede influir en la eficacia y la precisión en los datos.

Seguridad de datos: La tecnología utilizada para el proyecto debe garantizar la seguridad, integridad y confidencialidad de los datos.

3.9.5 Ámbito ecológico

Campañas con propósito: El lanzamiento de productos tuvo lugar en ubicaciones estratégicas, donde médicos y asistentes fueron informados sobre la relevancia de conservar este ecosistema natural. Se destacó la importancia de convivir con las comunidades locales, adquirir sus artesanías para contribuir al desarrollo económico y social. Además, se proporcionaron kits de salud a las comunidades como parte de esta iniciativa.

Medición de la huella de carbono: Se ha implementado planes preventivos y correctivos para reducir el impacto de C02 y contribuir a la mitigación del cambio climático. Se realizó alianzas estratégicas con una empresa recicladora con la finalidad de generar conciencia en los colaboradores de la compañía frente a la problemática de la contaminación por el mal manejo

de residuos, así como dignificar el trabajo de los recicladores base, a través de la división de desechos sólidos, como plástico, cartón y papel, desde la fuente.

Manejo responsable de los recursos: La empresa contribuyó a la reforestación con la siembra de plantas endémicas. La actividad contó con la participación de voluntarios de la institución y el apoyo de la Asociación de Scouts del Ecuador.

3.9.6 Ámbito legal

En Ecuador, el artículo 66, numeral 19 de la Constitución de la República, garantiza "el derecho a la salvaguarda de información de carácter personal, incluyendo el acceso y la toma de decisiones sobre datos de esta índole, así como su debida protección. La recopilación, almacenamiento, procesamiento, distribución o divulgación de dichos datos o información requieren la autorización del titular o el respaldo de la legislación correspondiente". Los artículos 25 a 51 de la Ley de Protección de Datos Personales detallan las diferentes categorías de datos, las condiciones para la transferencia o comunicación, el acceso a datos personales por parte de terceros, la seguridad de la información personal y la responsabilidad asociada a estos procesos.

Bajo esta ley se ha construido en el Laboratorio Farmacéutico una plataforma para poder recolectar el consentimiento, en donde el usuario autoriza a la empresa a tratar, almacenar y utilizar sus datos para Análisis y estadísticas demográficas sobre el uso de productos, entrega de premios en los casos aplicables. Realizar actividades de mercadotecnia, publicidad, telemarketing, promociones, la empresa se compromete resguardar y cumpliendo con la obligación de confidencialidad y previniendo razonablemente el uso o divulgación indebida de los mismos.

3.10 Planteamiento Agile

Para hacer el desarrollo de este proyecto se utilizará una metodología Scrum de la siguiente forma:

Ejecutar el sprint 0-planificación

Se inicia con el Sprint 0, el cual fue coordinado por el Jefe de proyecto (JP), quien planifica los plazos de entregas, actividades, responsables y recursos, etc. La documentación que aquí se genera se denomina Backlog y se ordena según prioridades. Adicional se definen los siguientes Roles:

- Ingeniero De Datos (ID)
- Arquitecto De Datos (AD)
- Científico De Datos (CD)
- Jefe De Proyecto (JP)

1. Ejecutar el sprint 1: Recopilación de datos

Durante el Sprint, el equipo trabajará en las siguientes historias de usuario (HU-006 hasta la HU-011), adicional se realizará reuniones diarias de seguimiento (Daily Scrum) para compartir avances y problemas.

a. **Revisar y Retroalimentar (Sprint Review)**: Al final del Sprint, el equipo presenta los resultados logrados durante el Sprint 1 al Jefe de Proyecto y a las partes interesadas. Se recopila retroalimentación y se discuten las lecciones aprendidas.
b. **Retrospectiva del Sprint (Sprint Retrospective):** El equipo lleva a cabo una sesión de retrospectiva con la finalidad de identificar áreas de mejora en cuanto a la

capacidad del equipo, la gestión del tiempo, y documenta exhaustivamente todo el proceso realizado. Durante este análisis, se destacan obstáculos y se proponen soluciones con el objetivo de optimizar el desarrollo del proceso.

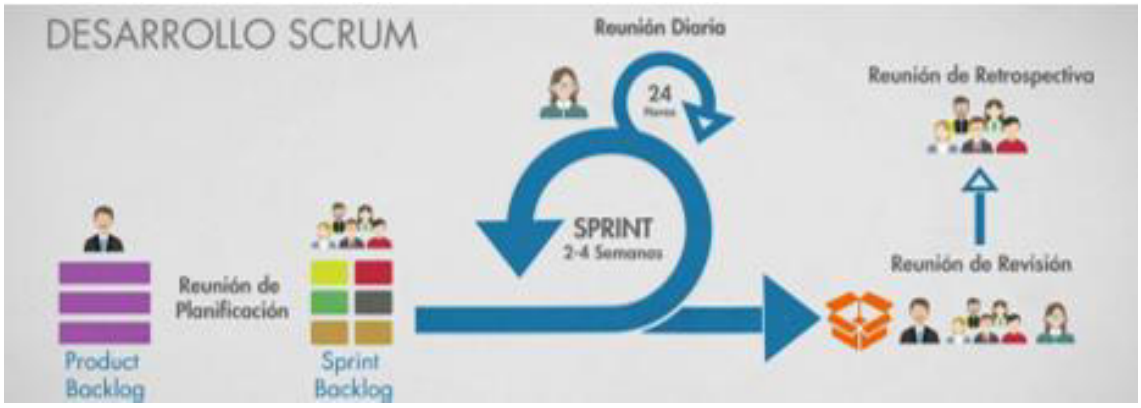
2. Ejecutar el sprint 2: Desarrollo del modelo

Durante el Sprint, el equipo trabajará en las siguientes historias de usuario (HU-012 hasta la HU-017), en donde se realizará de igual manera reuniones diarias de seguimiento, sprint review y sprint retrospective.

3. Ejecutar el sprint 3: Presentación del proyecto

Durante el Sprint, el equipo trabajará en las siguientes historias de usuario (HU-018 hasta la HU-019), en donde se realizará de igual manera reuniones diarias de seguimiento, sprint review y sprint retrospective.

Figura 8
Desarrollo de Scrum



Nota: Obtenido de https://www.iebschool.com/blog/wp-content/uploads/2021/04/como-funciona-scrum.png

Tabla 14
Metodología Scrum

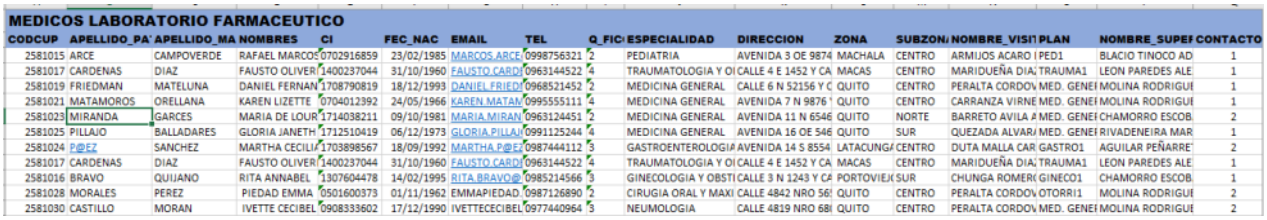
RESPONSABLE	*COD*	*Work Item*	*PRIORITY*	**TOTAL HORAS**
JP	**SPRINT-00**	**Fase 1: Planificación (Semana 1-4)**		70.00
JP	HU-001	Definición de Roles del equipo	5	10.00
	HU-002	Definición de objetivos y alcance del proyecto.	5	10.00
	HU-003	Identificación de los equipos y recursos necesarios.	5	10.00
	HU-004	Revisión de la literatura existente sobre detección de datos duplicados en PLN.	3	30.00
	HU-005	Establecimiento de métricas de evaluación de rendimiento.	5	10.00
ID-AD-CD-JP	**SPRINT-01**	**Fase 2: Recopilación de Datos (Semana 5-8)**		120.00
JP-CD	HU-006	Identificación de las fuentes de datos relevantes en el laboratorio farmacéutico.	5	30.00
AD	HU-007	Definir el flujo del proceso.	5	20.00
AD	HU-008	Elección del motor de base datos a utilizar.	5	10.00
AD	HU-009	Creación de una base de datos de entrenamiento y prueba.	5	20.00
ID	HU-010	Conexión de las bases de datos.	5	20.00
JP	HU-011	Elaboración de la documentación del avance realizado.	3	20.00
ID-AD-CD-JP	**SPRINT-02**	**Fase 3: Desarrollo del Modelo (Semana 9-12)**		120.00
ID-AD-CD	HU-012	Selección de algoritmos de NLP adecuados para la detección de duplicados.	5	40.00
CD	HU-014	Evaluación del rendimiento del modelo en datos de prueba.	4	20.00
ID-AD-CD	HU-015	Ajustes finales del modelo	4	30.00
JP	HU-016	Incorporación de los avances realizados a la documentación.	3	20.00
JP	**SPRINT-03**	**Fase 4: Presentación del proyecto(Semana 14-16)**		50.00
JP	HU-018	Revisión y ajuste del documento escrito.	5	30.00
JP	HU-019	Elaboración de la presentación y exposición del proyecto.	5	20.00
TOTAL DE HORAS				350.00

CAPITULO IV DESARROLLO

4.1 Archivo de recopilación de datos

En este proyecto se implementó un archivo Excel para la fase de recopilación de datos. Para reducir los errores de digitación en cuanto a la información de las columnas que manejan catálogos, se creó una macro programada en Visual Basic 6.0, la cual nos permite tener combo box en cada celda de cada una de las siguientes columnas catálogos: Especialidad, Zona, Subzona, Visitador, Planes, Supervisor.

Figura 9
Archivo de recopilación datos



MEDICOS LABORATORIO FARMACEUTICO

CODCUP	APELLIDO_PA	APELLIDO_MA	NOMBRES	CI	FEC_NAC	EMAIL	TEL	Q_FIC	ESPECIALIDAD	DIRECCION	ZONA	SUBZON	NOMBRE_VISI	PLAN	NOMBRE_SUPE	CONTACTO
2581015	ARCE	CAMPOVERDE	RAFAEL MARCOS	0702916859	23/02/1985	MARCOS.ARCE	0998756321	2	PEDIATRIA	AVENIDA 3 OE 9874	MACHALA	CENTRO	ARMIJOS ACARO I	PED1	BLACIO TINOCO AD	1
2581017	CARDENAS	DIAZ	FAUSTO OLIVER	1400237044	31/10/1960	FAUSTO.CARDI	0963144522	4	TRAUMATOLOGIA Y OI	CALLE 4 E 1452 Y CA	MACAS	CENTRO	MARIDUEÑA DIA	TRAUMA1	LEON PAREDES ALE	1
2581019	FRIEDMAN	MATELUNA	DANIEL FERNAN	1708790819	18/12/1993	DANIEL.FRIED	0968521452	2	MEDICINA GENERAL	CALLE 6 N 52156 Y C	QUITO	CENTRO	PERALTA CORDOV	MED. GENEI	MOLINA RODRIGUE	1
2581021	MATAMOROS	ORELLANA	KAREN LIZETTE	0704012392	24/05/1966	KAREN.MATAM	0995555111	4	MEDICINA GENERAL	AVENIDA 7 N 9876	QUITO	CENTRO	CARRANZA VIRNE	MED. GENEI	MOLINA RODRIGUE	1
2581023	MIRANDA	GARCES	MARIA DE LOUR	1714038211	09/10/1981	MARIA.MIRAN	0963124451	2	MEDICINA GENERAL	AVENIDA 11 N 6546	QUITO	NORTE	BARRETO AVILA A	MED. GENEI	CHAMORRO ESCOB	2
2581025	PILLAJO	BALLADARES	GLORIA JANETH	1712510419	06/12/1973	GLORIA.PILLAJ	0991125244	4	MEDICINA GENERAL	AVENIDA 16 OE 546	QUITO	SUR	QUEZADA ALVAR/	MED. GENEI	RIVADENEIRA MAR	1
2581024	P@EZ	SANCHEZ	MARTHA CECILIA	1703898567	18/09/1992	MARTHA.P@EZ	0987444112	3	GASTROENTEROLOGIA	AVENIDA 14 S 8554	LATACUNGA	CENTRO	DUTA MALLA CAR	GASTRO1	AGUILAR PEÑARRE	2
2581017	CARDENAS	DIAZ	FAUSTO OLIVER	1400237044	31/10/1960	FAUSTO.CARDI	0963144522	4	TRAUMATOLOGIA Y OI	CALLE 4 E 1452 Y CA	MACAS	CENTRO	MARIDUEÑA DIA	TRAUMA1	LEON PAREDES ALE	1
2581016	BRAVO	QUIJANO	RITA ANNABEL	1307604478	14/02/1995	RITA.BRAVO@	0985214566	3	GINECOLOGIA Y OBST	CALLE 3 N 1243 Y CA	PORTOVIEJO	SUR	CHUNGA ROMERO	GINECO1	CHAMORRO ESCOB	1
2581028	MORALES	PEREZ	PIEDAD EMMA	0501600373	01/11/1962	EMMAPIEDAD.	0987126890	2	CIRUGIA ORAL Y MAXI	CALLE 4842 NRO 56	QUITO	CENTRO	PERALTA CORDOV	OTORRI1	MOLINA RODRIGUE	2
2581030	CASTILLO	MORAN	IVETTE CECIBEL	0908333602	17/12/1990	IVETTECECIBEL	0977440964	3	NEUMOLOGIA	CALLE 4819 NRO 68	QUITO	CENTRO	PERALTA CORDOV	MED. GENEI	MOLINA RODRIGUE	2

Las funciones y métodos programados en la macro son los siguientes:

Figura 10
Código Visual Basic validación apellidos y nombres

```
Private Sub Worksheet_Change(ByVal Target As Range)
Application.ScreenUpdating = False
Application.DisplayStatusBar = True
Dim intNumeroFila As Integer
Dim intNumeroColumna As Integer
Dim Rng As Range
Dim Cell As Range
intNumeroFila = ActiveCell.Row
intNumeroColumna = ActiveCell.Column

Set Rng = Me.Columns("B")
If Not Intersect(Target, Rng) Is Nothing Then
    Application.EnableEvents = False
    For Each Cell In Target
        If Cell.Row <> 2 Then
            If Not EsTexto(Cell.Value) Then
                MsgBox "Solo se permiten letras en esta columna.", vbExclamation
                Cell.ClearContents
            End If
        End If
    Next Cell
    Application.EnableEvents = True
End If

Set Rng = Me.Columns("C")
If Not Intersect(Target, Rng) Is Nothing Then
    Application.EnableEvents = False
    For Each Cell In Target
        If Cell.Row <> 2 Then
            If Not EsTexto(Cell.Value) Then
                MsgBox "Solo se permiten letras en esta columna.", vbExclamation
                Cell.ClearContents
            End If
        End If
    Next Cell
    Application.EnableEvents = True
End If


Set Rng = Me.Columns("D")
If Not Intersect(Target, Rng) Is Nothing Then
    Application.EnableEvents = False
    For Each Cell In Target
        If Cell.Row <> 2 Then
            If Not EsTexto(Cell.Value) Then
                MsgBox "Solo se permiten letras en esta columna.", vbExclamation
                Cell.ClearContents
            End If
        End If
    Next Cell
    Application.EnableEvents = True
End If
```

Figura 11
Código Visual Basic validación cédula

```
Set Rng = Me.Columns("E")
' Verifica si los cambios se realizan en la columna deseada
If Not Intersect(Target, Rng) Is Nothing Then
    Application.EnableEvents = False ' Deshabilita temporalmente los eventos para evitar un bucle
    ' Recorre cada celda en la columna (excepto la fila 2) y verifica el contenido
    For Each Cell In Target
        If Cell.Row <> 2 Then ' Excluye la fila 2
            Valor = Cell.Value
            For i = 1 To Len(Valor)
                If Not EsCaracterPermitido(Mid(Valor, i, 1)) Then ' Verifica si el carácter no es permitido
                    MsgBox "Solo se permiten los caracteres 0-9 en esta columna.", vbExclamation
                    Cell.ClearContents ' Borra el contenido de la celda si contiene caracteres no permitidos
                    Exit For
                End If
            Next i
        End If
    Next Cell
    Application.EnableEvents = True ' Vuelve a habilitar los eventos
End If
```

Figura 12
Código Visual Basic validación fecha de nacimiento

```
Set Rng = Me.Columns("F")
If Not Intersect(Target, Rng) Is Nothing Then
    Application.EnableEvents = False
    For Each Cell In Target
        If Cell.Row <> 2 Then ' Excluye la fila 2
            If Not EsFecha(Cell.Value) Then ' Verifica si el valor no es una fecha
                MsgBox "Solo se permiten fechas en esta columna.", vbExclamation
                Cell.ClearContents ' Borra el contenido de la celda si no es una fecha
            End If
        End If
    Next Cell
    Application.EnableEvents = True ' Vuelve a habilitar los eventos
End If
```

Figura 13
Código Visual Basic validación celular

```
Set Rng = Me.Columns("H")
' Verifica si los cambios se realizan en la columna deseada
If Not Intersect(Target, Rng) Is Nothing Then
    Application.EnableEvents = False ' Deshabilita temporalmente los eventos para evitar un bucle
    ' Recorre cada celda en la columna (excepto la fila 2) y verifica el contenido
    For Each Cell In Target
        If Cell.Row <> 2 Then ' Excluye la fila 2
            Valor = Cell.Value
            For i = 1 To Len(Valor)
                If Not EsCaracterPermitido(Mid(Valor, i, 1)) Then ' Verifica si el carácter no es permitido
                    MsgBox "Solo se permiten los caracteres 0-9 en esta columna.", vbExclamation
                    Cell.ClearContents ' Borra el contenido de la celda si contiene caracteres no permitidos
                    Exit For
                End If
            Next i
        End If
    Next Cell
    Application.EnableEvents = True ' Vuelve a habilitar los eventos
End If
```

Figura 14
Código Visual Basic validación categoría y contactos(numéricos)

```
Set Rng = Me.Columns("I")
' Verifica si los cambios se realizan en la columna deseada
If Not Intersect(Target, Rng) Is Nothing Then
    Application.EnableEvents = False ' Deshabilita temporalmente los eventos para evitar un bucle
    ' Recorre cada celda en la columna (excepto la fila 2) y verifica el contenido
    For Each Cell In Target
        If Cell.Row <> 2 Then ' Excluye la fila 2
            Valor = Cell.Value
            For i = 1 To Len(Valor)
                If Not EsCaracterPermitido(Mid(Valor, i, 1)) Then ' Verifica si el carácter no es permitido
                    MsgBox "Solo se permiten los caracteres 0-9 en esta columna.", vbExclamation
                    Cell.ClearContents ' Borra el contenido de la celda si contiene caracteres no permitidos
                    Exit For
                End If
            Next i
        End If
    Next Cell
    Application.EnableEvents = True ' Vuelve a habilitar los eventos
End If

Set Rng = Me.Columns("Q")
' Verifica si los cambios se realizan en la columna deseada
If Not Intersect(Target, Rng) Is Nothing Then
    Application.EnableEvents = False ' Deshabilita temporalmente los eventos para evitar un bucle
    ' Recorre cada celda en la columna (excepto la fila 2) y verifica el contenido
    For Each Cell In Target
        If Cell.Row <> 2 Then ' Excluye la fila 2
            Valor = Cell.Value
            For i = 1 To Len(Valor)
                If Not EsCaracterPermitido(Mid(Valor, i, 1)) Then ' Verifica si el carácter no es permitido
                    MsgBox "Solo se permiten los caracteres 0-9 en esta columna.", vbExclamation
                    Cell.ClearContents ' Borra el contenido de la celda si contiene caracteres no permitidos
                    Exit For
                End If
            Next i
        End If
    Next Cell
    Application.EnableEvents = True ' Vuelve a habilitar los eventos
End If
```

Figura 15

Código Visual Basic validación con catálogo de especialidad

```
If (intNumeroColumna = 2 And intNumeroFila > 2) Then
    Cells(intNumeroFila - 1, intNumeroColumna + 8).Select
    With Selection.Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
        xlBetween, Formula1:="=Hoja2!$B$2:$B$44"
        .IgnoreBlank = True
        .InCellDropdown = True
        .InputTitle = ""
        .ErrorTitle = ""
        .InputMessage = ""
        .ErrorMessage = ""
        .ShowInput = True
        .ShowError = True
    End With
    Cells(intNumeroFila - 1, intNumeroColumna + 10).Select
    With Selection.Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
        xlBetween, Formula1:="=Hoja2!$D$2:$D$222"
        .IgnoreBlank = True
        .InCellDropdown = True
        .InputTitle = ""
        .ErrorTitle = ""
        .InputMessage = ""
        .ErrorMessage = ""
        .ShowInput = True
        .ShowError = True
    End With
    Cells(intNumeroFila - 1, intNumeroColumna + 11).Select
```

Figura 16

Código Visual Basic validación con catálogo de zona, subzona, visitador

```
Cells(intNumeroFila - 1, intNumeroColumna + 11).Select
With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
    xlBetween, Formula1:="=Hoja2!$F$2:$F$4"
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With
Cells(intNumeroFila - 1, intNumeroColumna + 12).Select
With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
    xlBetween, Formula1:="=Hoja2!$H$2:$H$101"
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With
Cells(intNumeroFila - 1, intNumeroColumna + 13).Select
With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
    xlBetween, Formula1:="=Hoja2!$J$2:$J$27"
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With
Cells(intNumeroFila - 1, intNumeroColumna + 14).Select
```

Figura 17
Código Visual Basic validación con catálogo supervisor

```
    With Selection.Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
        xlBetween, Formula1:="=Hoja2!$L$2:$L$21"
        .IgnoreBlank = True
        .InCellDropdown = True
        .InputTitle = ""
        .ErrorTitle = ""
        .InputMessage = ""
        .ErrorMessage = ""
        .ShowInput = True
        .ShowError = True
    End With
    Cells(intNumeroFila - 1, intNumeroColumna + 1).Select
ElseIf (intNumeroColumna = 3 And intNumeroFila > 2) Then
    Cells(intNumeroFila, intNumeroColumna + 7).Select
    With Selection.Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
        xlBetween, Formula1:="=Hoja2!$B$2:$B$44"
        .IgnoreBlank = True
        .InCellDropdown = True
        .InputTitle = ""
        .ErrorTitle = ""
        .InputMessage = ""
        .ErrorMessage = ""
        .ShowInput = True
        .ShowError = True
    End With
    Cells(intNumeroFila, intNumeroColumna + 9).Select
```

Figura 18
Código Visual Basic validación con catálogo de plan

```
With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
    xlBetween, Formula1:="=Hoja2!$D$2:$D$222"
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With
Cells(intNumeroFila, intNumeroColumna + 10).Select
With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
    xlBetween, Formula1:="=Hoja2!$F$2:$F$4"
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With
Cells(intNumeroFila, intNumeroColumna + 11).Select
With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
    xlBetween, Formula1:="=Hoja2!$N$2:$N$101"
    .IgnoreBlank = True
    .InCellDropdown = True
    .InputTitle = ""
    .ErrorTitle = ""
    .InputMessage = ""
    .ErrorMessage = ""
    .ShowInput = True
    .ShowError = True
End With
Cells(intNumeroFila, intNumeroColumna + 12).Select
```

Figura 19
Código Visual Basic validación de texto

```
        With Selection.Validation
            .Delete
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
            xlBetween, Formula1:="=Hoja2!$J$2:$J$27"
            .IgnoreBlank = True
            .InCellDropdown = True
            .InputTitle = ""
            .ErrorTitle = ""
            .InputMessage = ""
            .ErrorMessage = ""
            .ShowInput = True
            .ShowError = True
        End With
        Cells(intNumeroFila, intNumeroColumna + 13).Select
        With Selection.Validation
            .Delete
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
            xlBetween, Formula1:="=Hoja2!$L$2:$L$21"
            .IgnoreBlank = True
            .InCellDropdown = True
            .InputTitle = ""
            .ErrorTitle = ""
            .InputMessage = ""
            .ErrorMessage = ""
            .ShowInput = True
            .ShowError = True
        End With
        Cells(intNumeroFila, intNumeroColumna).Select
    End If
    Application.ScreenUpdating = True
    End Sub

    Function EsTexto(ByVal Valor As Variant) As Boolean
        If IsEmpty(Valor) Then
            EsTexto = True
        ElseIf IsString(Valor) Then
            EsTexto = True
        Else
            EsTexto = False
        End If
    End Function
```

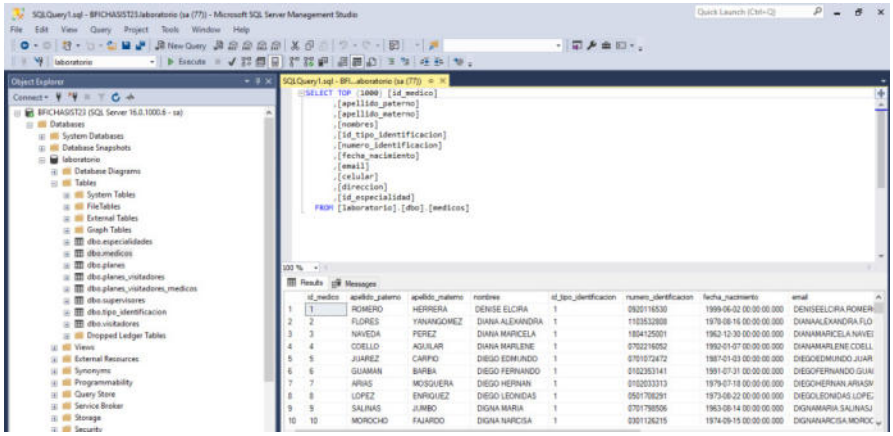
Figura 20
Código Visual Basic validación campos de fecha y numéricos

```
Function IsString(ByVal Valor As Variant) As Boolean
    ' Función que verifica si el valor es una cadena de texto
    If TypeName(Valor) = "String" Then
        IsString = True ' Si el valor es una cadena de texto, se considera una cadena de texto
    Else
        IsString = False ' En otros casos, no se considera una cadena de texto
    End If
End Function


Function EsFecha(ByVal Valor As Variant) As Boolean
    ' Función que verifica si el valor es una fecha
    If IsEmpty(Valor) Then
        EsFecha = True
    ElseIf IsDate(Valor) Then
        EsFecha = True ' Si el valor es una fecha válida, se considera una fecha
    Else
        EsFecha = False ' En otros casos, no se considera una fecha
    End If
End Function

Function EsCaracterPermitido(ByVal Caracter As String) As Boolean
    ' Función que verifica si el carácter es permitido (0-9)
    Select Case Caracter
        Case "0" To "9"
            EsCaracterPermitido = True ' Si el carácter es un dígito, se considera permitido
        Case Else
            EsCaracterPermitido = False ' En otros casos, no se considera permitido
    End Select
End Function
```

4.2 Implementación de ambiente de desarrollo y test

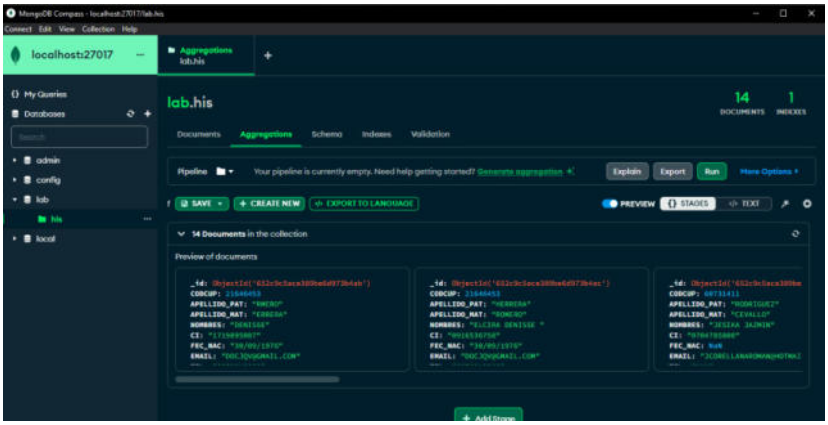
Para el desarrollo de este proyecto, se levantó de manera local un servidor de SQL Server v16.0 para el almacenamiento de la información de médicos nuevos.

Figura 21
Ambiente de desarrollo y test SQL Server



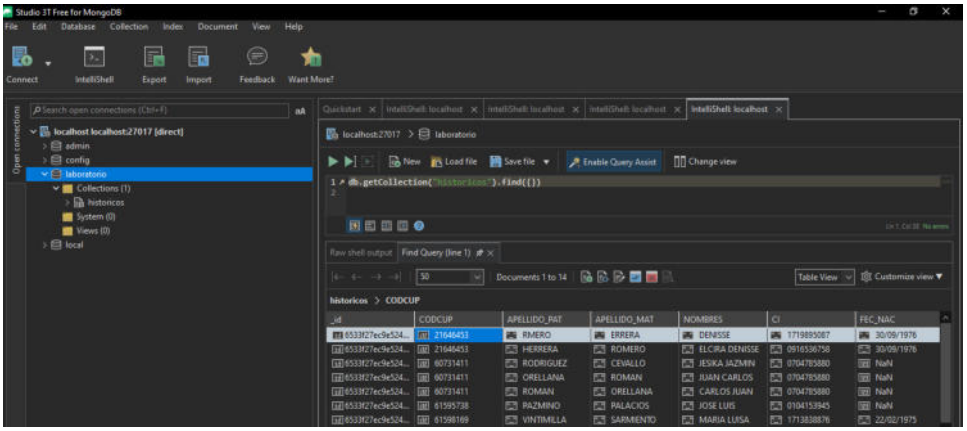
Adicionalmente, se levantó de manera local un servidor de MongoDB para almacenar la información de los médicos tanto nuevos como duplicados y de esta forma tener una tabla de históricos de la información.

Figura 22
Servidor Mongo DB históricos



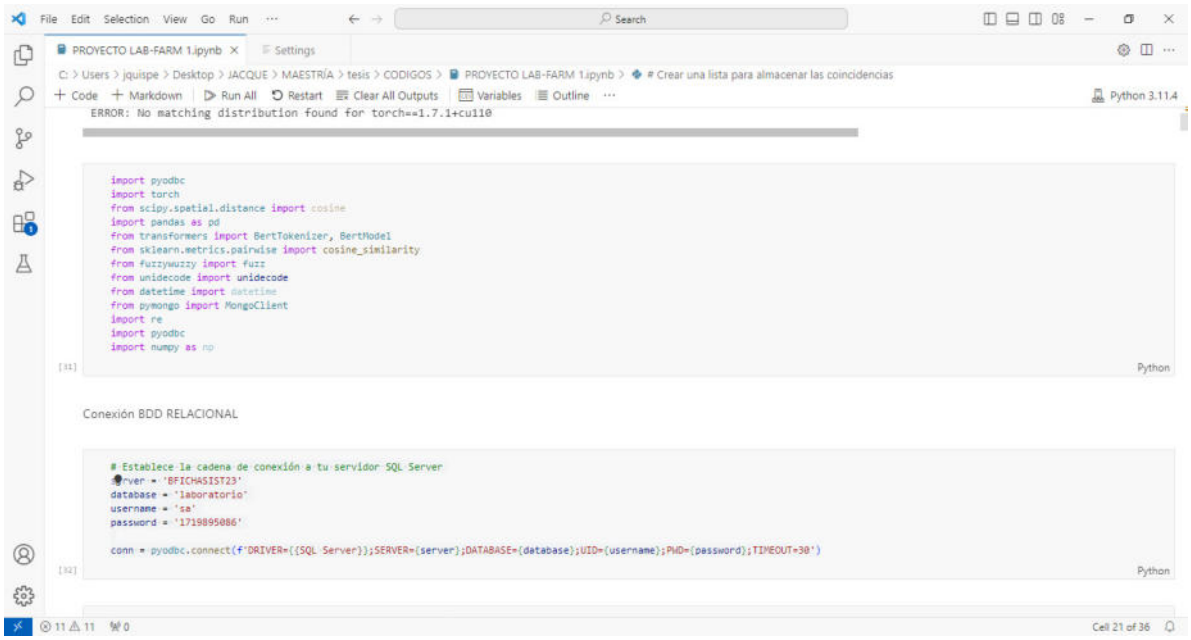
Para interactuar con la base MongoDB se utilizó la herramienta Studio 3T v2023.5.0

Figura 23
Ambiente de consultas Mongo DB



El desarrollo del modelo se lo realizo en lenguaje Python v3.11.0 y como editor de código se utilizó la aplicación Visual Studio Code v1.83.1

Figura 24
Programación en VSC

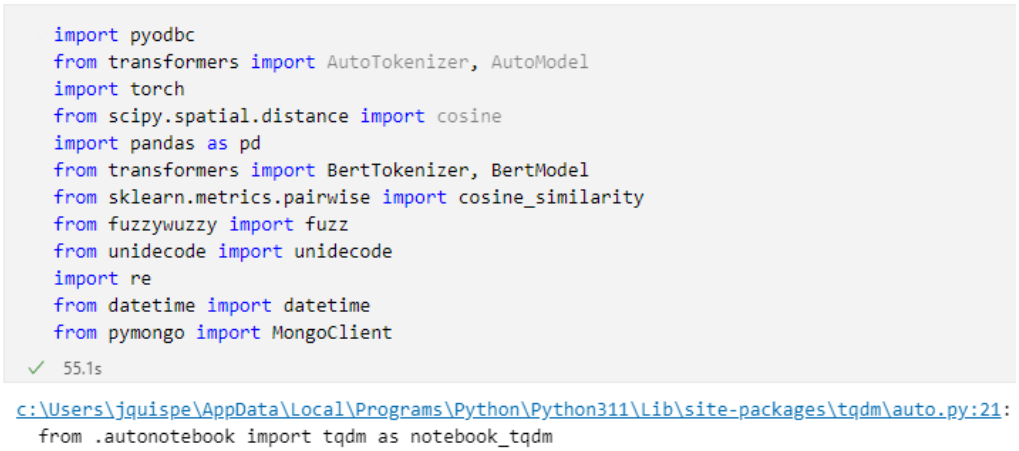


4.3 Programación del proyecto

4.3.1 Importación De Bibliotecas

Para llevar a cabo diversas tareas relacionadas con el procesamiento de datos, el procesamiento del lenguaje natural, cálculos de similitud y manipulación de bases de datos, entre otros, se han instalado las librerías necesarias detallas a continuación:

Figura 25
Importación de bibliotecas



4.3.2 Comando ejecutado para la conexión con la BDD de SQL SERVER

En esta sección del proyecto, se ha configurado la conexión con el servidor SQL Server para acceder a la base de datos que almacena la información de médicos del laboratorio farmacéutico. La conexión se establece mediante la biblioteca PYODBC.

Los componentes utilizados para la conexión son Server (BFICHASIST23) que especifica el nombre del servidor al que se está conectando, en este caso.

Database (laboratorio) Indica el nombre de la base de datos específica a la que se accederá.

username y password: Proporcionan las credenciales de inicio de sesión necesarias para autenticar y autorizar el acceso a la base de datos. En este caso, el usuario 'sa' se utiliza junto con la contraseña correspondiente '171989508'.

pyodbc.connect: Se utiliza para establecer la conexión con el servidor SQL Server mediante la especificación del controlador ('SQL Server') y la concatenación de los parámetros de conexión, incluidos el servidor, la base de datos, el nombre de usuario y la contraseña.

Figura 26
Conexión BBD SQL Server

```
# Establece la cadena de conexión a tu servidor SQL Server
server = 'BFICHASIST23'
database = 'laboratorio'
username = 'sa'
password = '1719895086'

conn = pyodbc.connect(f'DRIVER={{SQL Server}};SERVER={server};DATABASE={database};UID={username};PWD={password};TIMEOUT=30')
```

33.7s Python

4.3.2.1 Consulta de base de datos de médicos

En este código se ha realizado una consulta a la base de datos utilizando un query SQL para recuperar los siguientes campos de la tabla 'medicos': 'id_medico', 'apellido_paterno', 'apellido_materno', 'nombres', 'numero_identificacion' e 'id_especialidad'. Los resultados de la consulta se almacenan en un DataFrame utilizando la función pd.read_sql_query. Esto facilita el manejo y el análisis de los datos recuperados en un formato tabular, lo que es esencial para la posterior comparación y detección de duplicados en nombres de médicos.

Figura 27
Query base médicos

```
# Ejecutamos el query para realizar la consulta a la base
base_med = "SELECT id_medico, apellido_paterno, apellido_materno, nombres, numero_identificacion, id_especialidad FROM dbo.medicos"

#  Se guarda en un df
sql_data = pd.read_sql_query(base_med, conn)
```

4.3.2.2 Consulta de Especialidades Médicas

Esta consulta nos permite acceder a todos los campos de la tabla 'especialidades', lo que proporciona una visión integral de todas las especialidades médicas registradas en la base de datos del laboratorio farmacéutico.

Figura 28
Query base de especialidades

```
# Ejecutamos el query para realizar la consulta a la base
especialidad = "select id_especialidad ID_ESPECIALIDAD, nombre_especialidad ESPECIALIDAD from dbo.especialidades"
# Se guarda en un df
especialidad = pd.read_sql_query(especialidad, conn)
especialidad.head()
```

4.3.3 Carga Del Archivo Excel

La carga de información nueva enviada por parte de los supervisores se cargará mediante el siguiente código, el cual nos permite cargar el archivo Excel con macros e iterar desde la segunda fila añadiendo al código skiprows = 1.

Figura 29
Código para carga de archivo Excel

```
# Lee el archivo Excel en un DataFrame
med_nuevos = pd.read_excel('C:/Users/jquispe/Desktop/JACQUE/MAESTRÍA/tesis/NLP/INSUMO/ALTAS MED.xlsm', skiprows=1)
#Para no afectar la tabla inicial generamos una copia
med_nuevos2 = med_nuevos
```

Se selecciona las columnas con las cuales se va a trabajar en el modelo y tratamiento de datos.

Figura 30
Código para seleccionar las columnas a trabajar

```
#Seleccionamos las columnas con las que se va a trabajar
med_nuevos2 = med_nuevos2.iloc[:, :-6].copy()
```

Se realiza la concatenacion de los campos de nombres y apellidos para poder ejecutar el modelo.

Figura 31
Código para concatenar los campos

```
#Se concatena el nombre y apellidos para poder ejecutar el modelo
med_nuevos2['NOMBRE_COMPLETO_EXC'] = med_nuevos2['APELLIDO_PAT'] + ' ' + med_nuevos2['APELLIDO_MAT'] + ' ' + med_nuevos2['NOMBRES']
```

Con la finalidad de homologar la información mantenida en las bases vs la información nueva se procede a asignar el codigo de la especialidad.

Figura 32
Merge con base de especialidades

```
# Realiza la fusión (merge) de los DataFrames para obtener el código de especialidad
med_nuevos2 = med_nuevos2.merge(especialidad, on='ESPECIALIDAD', how='left')
# Renombra la columna 'Codigo' a 'Especialidad'
med_nuevos2 = med_nuevos2.rename(columns={'id_especialidad': 'cod_Espe'})
# Eliminar una columna específica del DataFrame
med_nuevos2 = med_nuevos2.drop('ESPECIALIDAD', axis=1)
```

Finalmente se visualiza la información cargada

Figura 33
Visualización del archivo cargado

```
med_nuevos2.head()
```

Python

```
c:\Users\jguisge\AppData\Local\Programs\Python\Python311\Lib\site-packages\openpyxl\worksheet\_reader.py:329: UserWarning: Data Validation extension is not supported and will be removed
  warn(msg)
```

	CODCUP	APELLIDO_PAT	APELLIDO_MAT	NOMBRES	CI	FEC_NAC	EMAIL	TEL	Q_FICO	ESPECIALIDAD	DIRECCION	NOMBRE_COMPLETO_EXC
0	2581015	ARCE	CAMPOVERDE	RAFAEL MARCOS	702916859	1985-02-23	MARCOS.ARCE@GMAIL.COM	998756321	2	PEDIATRÍA	AVENIDA 3 OE 9874 Y CALLE 1	ARCE CAMPOVERDE RAFAEL MARCOS
1	2581017	CARDENAS	DIAZ	FAUSTO OLIVERIO	1400237044	1960-10-31	FAUSTO.CARDENAS@GMAIL.COM	963144522	4	TRAUMATOLOGÍA Y ORTOPEDIA	CALLE 4 E 1452 Y CALLE 8	CARDENAS DIAZ FAUSTO OLIVERIO
2	2581019	FRIEDMAN	MATELUNA	DANIEL FERNANDO	1708790819	1993-12-18	DANIEL.FRIEDMAN@GMAIL.COM	968521452	2	MEDICINA GENERAL	CALLE 6 N 52156 Y CALLE 11	FRIEDMAN MATELUNA DANIEL FERNANDO
3	2581021	MATAMOROS	ORELLANA	KAREN LIZETTE	704012392	1966-05-24	KAREN.MATAMOROS@GMAIL.COM	995555111	4	MEDICINA GENERAL	AVENIDA 7 N 9876 Y AVENIDA 65	MATAMOROS ORELLANA KAREN LIZETTE
4	2581023	MIRANDA	GARCES	MARIA DE LOURDES	1714038211	1981-10-09	MARIA.MIRANDA@GMAIL.COM	963124451	2	MEDICINA GENERAL	AVENIDA 11 N 6546 Y CALLE 5	MIRANDA GARCES MARIA DE LOURDES

4.3.4 Pre procesamiento de los datos

La fase de pre procesamiento de los datos es bastante importante para garantizar la calidad, integridad y consistencia de los datos, por tal motivo se ejecutó una serie de comandos que nos permitió que los datos estén limpios, coherentes y listos para ser utilizados en la etapa de construcción del modelo de detección de datos duplicados basado en NLP, lo que contribuye a la eficiencia y la precisión del modelo.

Como primer paso se valida el tipo de datos que se tiene en la dataframe.

Figura 34
Código de visualización de tipo de datos del archivo excel

```
med_nuevos2.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   CODCUP               12 non-null     int64
 1   APELLIDO_PAT         12 non-null     object
 2   APELLIDO_MAT         12 non-null     object
 3   NOMBRES              12 non-null     object
 4   CI                   12 non-null     int64
 5   FEC_NAC              12 non-null     datetime64[ns]
 6   EMAIL                12 non-null     object
 7   TEL                  12 non-null     int64
 8   Q_FICO               12 non-null     int64
 9   DIRECCION            12 non-null     object
 10  NOMBRE_COMPLETO_EXC  12 non-null     object
 11  ID_ESPECIALIDAD      6 non-null      float64
dtypes: datetime64[ns](1), float64(1), int64(4), object(6)
memory usage: 1.3+ KB
```

4.3.4.1 Eliminación de Tildes y Conversión de "ñ"

Para asegurar que los datos de texto estén en un formato uniforme, se aplicó una función que eliminó las tildes de las letras acentuadas y convirtió la "ñ" en "n" en todas las columnas de tipo texto. Esto garantiza que no haya variaciones debidas a acentos o caracteres especiales que puedan dificultar la detección de duplicados, así como también asegura que la información que se cargue a la base no vaya con caracteres especiales difíciles de leer o que dificulten la eficiencia del modelo.

Figura 35
Código para eliminación de tildes y Ñ

```
# Función para eliminar tildes y convertir "ñ" en "n" en una cadena
def quitar_tildes_y_n(cadena):
    if isinstance(cadena, str):
        cadena_sin_tildes = unidecode(cadena)
        cadena_sin_n = cadena_sin_tildes.replace('ñ', 'n')
        return cadena_sin_n
    else:
        return cadena

# Aplicar la función eliminar tildes y ñ solo a las columnas de tipo objeto (texto)
for columna in med_nuevos2.select_dtypes(include=['object']).columns:
    med_nuevos2[columna] = med_nuevos2[columna].apply(quitar_tildes_y_n) # esta funcion quita las tildes y las ñ
```

4.3.4.2 Eliminación de Espacios en Blanco

Se procedió a eliminar espacios en blanco al principio y al final de los valores de las columnas de texto, lo que contribuye a la consistencia de los datos.

Figura 36
Código para eliminación de espacios en blanco

```
# Eliminación de espacios en blanco
med_nuevos2[columna] = med_nuevos2[columna].str.strip()
```

4.3.4.3 Eliminación de Filas Duplicadas

Con el objetivo de evitar redundancias, se eliminaron las filas duplicadas en el conjunto de datos.

Figura 37
Código para eliminación de filas duplicadas

```
# elimina las filas duplicadas
med_nuevos2 = med_nuevos2.drop_duplicates()
```

4.3.4.4 Normalización de Datos Numéricos como Texto

Algunas columnas que originalmente contenían datos numéricos, como "TEL" y "Q_FICO," se convirtieron a tipo texto para asegurar que tengan la estructura que requiere la BBD del laboratorio farmacéutico. Adicional, se eliminaron los decimales ".0".

Figura 38
Código para conversión a tipo texto

```
# Convierte en tipo texto
med_nuevos2['TEL'] = med_nuevos2['TEL'].apply(lambda x: str(x).replace('.0', ''))
med_nuevos2['Q_FICO'] = med_nuevos2['Q_FICO'].apply(lambda x: str(x).replace('.0', ''))
med_nuevos2['CI'] = med_nuevos2['CI'].astype(str)
```

4.3.4.5 Estandarización de Formatos de Cédula y Número de Teléfono

Se aplicó una función específica para que tanto los números de cédula como los números de teléfono tengan una estructura de 10 dígitos. Si el número de cédula o celular estan compuesto por solo 9 dígitos, se le agregó un "0" al principio.

Figura 39
Código para conversión a tipo texto

```python
# Función para que la cedula y celular tenga la estructura de 10 digitos
def convertir_a_texto(valor):
    if isinstance(valor, str):
        valor_texto = str(valor)
        if len(valor_texto) == 9:
            valor_texto = '0' + valor_texto
        return valor_texto
    else:
        return str(valor)

# Aplicar la función solo a las columnas "ci" y "celular"
med_nuevos2['TEL'] = med_nuevos2['TEL'].apply(convertir_a_texto)
med_nuevos2['CI'] = med_nuevos2['CI'].apply(convertir_a_texto)
```

4.3.4.6 Eliminación de Caracteres Especiales

Para garantizar que los datos de texto sean coherentes y puedan ser procesados de manera efectiva, se eliminaron todos los caracteres especiales de las columnas de texto, a excepción de la columna "EMAIL".

Figura 40
Código para eliminación de caracteres especiales

```python
# Función para eliminar caracteres especiales de una cadena
def eliminar_caracteres_especiales(cadena):
    if isinstance(cadena, str):
        return re.sub(r'[^\w\s]', '', cadena)
    else:
        return cadena

# Obtener las columnas de texto (excepto la columna de correo)
columnas_texto = [col for col in med_nuevos2.select_dtypes(include=['object']).columns if col != 'EMAIL']

# Iterar a través de las columnas de texto y aplicar la función
for columna in columnas_texto:
    med_nuevos2[columna] = med_nuevos2[columna].apply(eliminar_caracteres_especiales)
```

4.3.4.7 Conversión a mayúsculas

Con el fin de homogeneizar el formato, todo el texto presente en el conjunto de datos se convirtió a mayúsculas.

Figura 41
Código para conversión a mayúsculas

```python
# convierte en mayusculas absolutamente todo el texto
med_nuevos2 = med_nuevos2.applymap(lambda x: x.upper() if isinstance(x, str) else x)
```

4.3.4.8 Conversión de la fecha de nacimiento a texto con formato adecuado

Para preparar los datos de fecha de nacimiento que se cargarán en la base de manera consistente y compatible, se transformó la columna "FEC_NAC". Para que cumpla con los requisitos de la base y faciliten la gestión de datos.

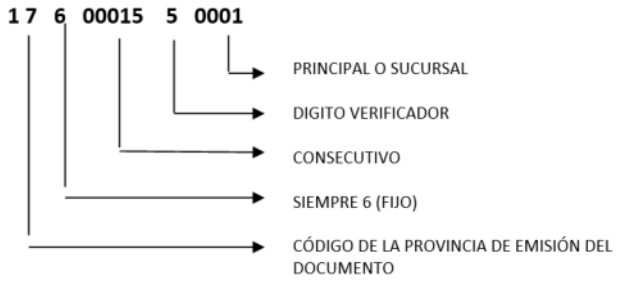
Figura 42
Código para conversión a fecha

```
#Fecha de nacimiento a texto y con el formato adecuado para cargar a las bases
med_nuevos2['FEC_NAC'] = med_nuevos2['FEC_NAC'].dt.strftime('%d/%m/%Y')
```

4.3.4.9 Algoritmo verificador de cédulas

La cédula es un identificador importante en el contexto de la gestión de datos en un laboratorio farmacéutico, y es fundamental garantizar que los números de cédula sean válidos y cumplan con los estándares establecidos. Debido a ello se generó la siguiente función para verificar la validez de los números de cédula presentes en la columna "CI" del conjunto de datos.

Figura 43
Descripción de algoritmo verificador de cédulas



Nota: Obtenido de https://miro.medium.com/v2/resize:fit:720/format:webp/1*b5UGr1eysFd9s6cbX1mNlg.png

Esta función verifica primero que la cédula este compuesta por 10 dígitos, comprueba que los primeros dos dígitos representen una provincia válida en Ecuador (números del 01 al 24). Valida el tercer dígito, conocido como "dígito de control," que debe estar en el rango de 0 a 5.

Posterior a ello calcula el "dígito de verificación" utilizando un algoritmo específico, que involucra la multiplicación de los dígitos de la cédula por coeficientes específicos y la suma de los productos resultantes sería el digito verificador.

Compara el "dígito verificador" calculado con el proporcionado en la cédula y verifica si coinciden. En caso de coincidencia, la función devuelve True, lo que indica que la cédula es válida; de lo contrario, devuelve False.

Figura 44
Código para validación de cédula de identidad

```
def validar_cedula(cedula):
    # Verificar que la cédula tenga 10 dígitos
    if len(cedula) != 10:
        return False

    # Verificar que los primeros dos dígitos sean válidos
    provincia = int(cedula[:2])
    if provincia < 1 or provincia > 24:
        return False

    # Verificar el tercer dígito (dígito de control)
    tercer_digito = int(cedula[2])
    if tercer_digito < 0 or tercer_digito > 5:
        return False

    # Verificar el último dígito (dígito de verificación)
    verificacion = int(cedula[9])

    # Calcular el dígito de verificación esperado
    coeficientes = [2, 1, 2, 1, 2, 1, 2, 1, 2]
    suma = 0
    for i in range(9):
        digito = int(cedula[i])
        producto = digito * coeficientes[i]
        if producto > 9:
            producto -= 9
        suma += producto

    verificacion_esperada = 10 - (suma % 10)
    if verificacion_esperada == 10:
        verificacion_esperada = 0

    # Comparar el dígito de verificación calculado con el proporcionado
    return verificacion == verificacion_esperada

# Aplicacion de la funcion
med_nuevos2['verif_ced'] = med_nuevos2['CI'].apply(validar_cedula)
med_nuevos2
```

4.3.5 Modelo de detección de duplicados

Para cumplir con el objetivo del presente proyecto se lleva a cabo una comparación entre los nombres de médicos presentes en un archivo Excel y los nombres almacenados en la base de

datos. Su finalidad es identificar posibles duplicados o similitudes significativas entre los nombres de médicos. A través de los siguientes modelos:

4.3.5.1 Modelo Fuzzywuzzy

Dentro de este modelo existen diferentes algoritmos que nos permite tener una detección exitosa de datos duplicados, para la ejecución de cada uno de ellos se realizó el siguiente procedimiento:

Creación de una Lista de Coincidencias: Se inicializa una lista llamada coincidencias que se utilizará para almacenar las coincidencias identificadas entre los nombres del archivo Excel y los nombres de la base de datos.

Figura 45
Código para creas listas

```
# Crear una lista para almacenar las coincidencias
coincidencias = []
```

Iteración a través de los nombres: Se itera a través de los nombres presentes en el archivo Excel y se comparan con los nombres almacenados en la base de datos SQL. Se utiliza la función de similitud de texto de la biblioteca fuzzywuzzy para calcular la similitud entre los nombres y apellidos.

Figura 46
Código para iteración a través de las filas

```
# Iterar a través de los nombres en el archivo Excel
for index_excel, row_excel in med_nuevos2.iterrows():
    nombre_excel = row_excel['nombre_completo_exc']
    mejor_coincidencia = None
    mejor_porcentaje = 0

    for index_db, row_db in sql_data2.iterrows():
        nombre_db = row_db['nombre_completo_sql']
```

Identificación de la mejor coincidencia: Se identifica la mejor coincidencia en función del porcentaje de similitud más alto. Si el porcentaje de similitud calculado supera un umbral predefinido, se considera una coincidencia significativa.

Figura 47
Código para identificación mejor coincidencia

```
# Si la similitud está por encima del umbral y es mejor que la anterior, actualizar la mejor coincidencia
if similitud_nombre > mejor_porcentaje:
    mejor_coincidencia = pd.concat([row_excel, row_db])
    mejor_coincidencia['Porcentaje de Similitud'] = similitud_nombre
    mejor_porcentaje = similitud_nombre
```

Almacenamiento de la mejor coincidencia: Se almacena la mejor coincidencia encontrada en la lista coincidencias, que contendrá tanto la información del archivo Excel como la información de la base de datos correspondiente.

Figura 48
Código para almacenamiento de coincidencias

```
# Agregar la mejor coincidencia a la lista de coincidencias
if mejor_coincidencia is not None:
    coincidencias.append(mejor_coincidencia)
```

Creación de un DataFrame de coincidencias: Se crea un DataFrame llamado coincidencias_df utilizando la lista de coincidencias, que proporciona una visión clara de todas las coincidencias identificadas entre los nombres de los médicos en el archivo Excel y los nombres en la base de datos SQL.

Figura 49
Código para creación de dataframe

```
# Crear un DataFrame con las mejores coincidencias
coincidencias_df = pd.DataFrame(coincidencias)
```

Asignación de Etiquetas de Observación: Se define una función asignar_medico que asigna etiquetas de "MEDICO EN BASE" o "MEDICO NUEVO" en función del porcentaje de similitud calculado. Esto ayuda a distinguir entre los médicos existentes en la base de datos y los médicos nuevos identificados en el archivo Excel.

Figura 50
Código para asignación de etiquetas

```
# Función para asignar "Medico en Base" o "Medico Nuevo" en función de la similitud
def asignar_medico(row):
    return "MEDICO EN BASE" if row['Porcentaje de Similitud'] > 79 else "MEDICO NUEVO"

# Aplicar la función a cada fila y crear una nueva columna "OBSERVACION"
coincidencias_df['OBSERVACION'] = coincidencias_df.apply(asignar_medico, axis=1)
```

Validación de cédulas: Con la finalidad de verificar que la cédula del médico encontrado en la base y el médico enviado en el archivo Excel corresponde a la misma persona se realiza una validación de cédulas de identidad, en el caso que el modelo de como resultado que el médico está en la base, en el campo de val_cedula debería dar un valor de ***True,*** caso contrario ***False***.

Figura 51
Validación cédula base vs archivo

```
# Validación de cédulas
coincidencias_df['VAL_CEDULA'] = coincidencias_df['CI'] == coincidencias_df['numero_identificacion']
```

Validación de especialidad: Al igual que la validación de cédula, se requiere verificar si la especialidad del médico encontrado en la base y el médico enviado en el archivo Excel corresponde a la misma persona se realiza la respectiva validación, en el caso que el modelo de como resultado que el médico está en la base, en el campo de val_especialidad debería dar un valor de ***True,*** caso contrario ***False***.

Figura 52
Validación de especialidad base vs archivo

```
# Validación de especialidad
coincidencias_df['VAL_ESPECIALIDAD'] = coincidencias_df['ID_ESPECIALIDAD'] == coincidencias_df['id_especialidad']
```

4.3.5.2 Algoritmo Fuzzy-Simple Ratio

En el caso de este algoritmo la función que se debe llamar para que nos permita calcular la similitud es:

Figura 53
Código para cálculo de similitud fuzzy-simple ratio

```
    # Calcular la similitud entre los nombres y apellidos
    similitud_nombre = fuzz.ratio(nombre_excel.lower(), nombre_db.lower())
```

En donde se obtuvo los siguientes resultados:

Figura 54
Código para visualización de resultados fuzzy-simple ratio

ODCUP	CI	NOMBRE_COMPLETO_EXC	ID_ESPECIALIDAD	verif_ced	id_medico	numero_identificacion	id_especialidad	NOMBRE_COMPLETO_SQL	Porcentaje de Similitud	OBSERVACION	VAL_CED	VAL_ESPE
81014	0702884826	ALVARADO CORDOVA MARCY RODELY	23	True	14	0906016357	34	ALVARADO CEDEÑO DORA VILMA	61	MEDICO NUEVO	False	False
81016	1307604478	BRAVO QUIJANO RITA ANNABEL	15	True	3122	1307604478	15	BRAVO QUIJANO RITA ANNABEL	100	MEDICO EN BASE	True	True
81018	1103175459	CORDOVA PALADINES JENNY ELIZABETH	37	True	2232	1001731213	7	DIAZ AYALA GENNY ELIZABETH	70	MEDICO NUEVO	False	False
81020	1709036824	LEMA LEMA WILSON ENRIQUE	22	True	1502	0701317778	39	VILELA SILVA JULIO ENRIQUE	67	MEDICO NUEVO	False	False
81022	1804027710	MEDINA DIAZ ANABEL EMMANUELLA	23	True	442	1802266435	1	GARCIA MEDINA SANDRA ELENA	57	MEDICO NUEVO	False	False
81024	1703898567	PAEZ SANCHEZ MARTHA CECILIA	12	True	3121	1703898567	12	PEZ SANCHEZ MARTHA CECILIA	98	MEDICO EN BASE	True	True
81026	1307584522	PIN GUADAMUD JINMY ALEXANDER	15	True	2781	0104789078	21	DELEG GUAZHA MAYRA ALEXANDRA	60	MEDICO NUEVO	False	False
81020	1709036824	LEMA LEMA WILSON ENRIQUE	23	True	1502	0701317778	39	VILELA SILVA JULIO ENRIQUE	67	MEDICO NUEVO	False	False
81027	0701791634	REDROVAN TENESACA EDITA FRANCISCA	20	True	29	0701791634	20	REDROVAN TENESACA EDITA FRANCISCA	99	MEDICO EN BASE	True	True
81029	0913837910	SALAS DAU FLORA SORAYA	16	True	68	0913837910	16	SALAS DAU FLORA SORAYA	100	MEDICO EN BASE	True	True
81031	0104262316	MALDONADO CABRERA MANUEL EFREN	5	True	251	0104262316	5	MALDONADO CABRERA MANUEL EFREN	100	MEDICO EN BASE	True	True

4.3.5.3 Algoritmo Fuzzy – Partial Ratio

La biblioteca utilizada para este algoritmo es:

Figura 55
Código para cálculo de similitud fuzzy-partial ratio

```
    # Calcular la similitud entre los nombres y apellidos
    similitud_nombre = fuzz.partial_ratio(nombre_excel.lower(), nombre_db.lower())
```

En donde se obtuvo los siguientes resultados:

Figura 56
Código para visualización de resultados fuzzy-partial ratio

CODCUP	CI	NOMBRE_COMPLETO_EXC	ID_ESPECIALIDAD	verif_ced	id_medico	numero_identificacion	id_especialidad	NOMBRE_COMPLETO_SQL	Porcentaje de Similitud	OBSERVACION	VAL_CED	VAL_ESP
2581014	0702884826	ALVARADO CORDOVA MARCY RODELY	23	True	14	0906016357	34	ALVARADO CEDEÑO DORA VILMA	63	MEDICO NUEVO	False	False
2581016	1307604478	BRAVO QUIJANO RITA ANNABEL	15	True	3122	1307604478	15	BRAVO QUIJANO RITA ANNABEL	100	MEDICO EN BASE	True	True
2581018	1103175459	CORDOVA PALADINES JENNY ELIZABETH	37	True	2232	1001731213	7	DIAZ AYALA GENNY ELIZABETH	74	MEDICO NUEVO	False	False
2581020	1709036824	LEMA LEMA WILSON ENRIQUE	22	True	1502	0701317778	39	VILELA SILVA JULIO ENRIQUE	68	MEDICO NUEVO	False	False
2581022	1804027710	MEDINA DIAZ ANABEL EMMANUELLA	23	True	911	1600194854	13	DIAZ SANCHEZ MARIA ANGELICA	60	MEDICO NUEVO	False	False
2581024	1703898567	PAEZ SANCHEZ MARTHA CECILIA	12	True	3121	1703898567	12	PEZ SANCHEZ MARTHA CECILIA	96	MEDICO EN BASE	True	True
2581026	1307584522	PIN GUADAMUD JINMY ALEXANDER	15	True	2781	0104789078	21	DELEG GUAZHA MAYRA ALEXANDRA	63	MEDICO NUEVO	False	False
2581020	1709036824	LEMA LEMA WILSON ENRIQUE	23	True	1502	0701317778	39	VILELA SILVA JULIO ENRIQUE	68	MEDICO NUEVO	False	False
2581027	0701791634	REDROVAN TENESACA EDITA FRANCISCA	20	True	29	0701791634	20	REDROVAN TENESACA EDITA FRANCISCA	97	MEDICO EN BASE	True	True
2581029	0913837910	SALAS DAU FLORA SORAYA	16	True	68	0913837910	16	SALAS DAU FLORA SORAYA	100	MEDICO EN BASE	True	True
2581031	0104262316	MALDONADO CABRERA MANUEL EFREN	5	True	251	0104262316	5	MALDONADO CABRERA MANUEL EFREN	100	MEDICO EN BASE	True	True

4.3.5.4 Algoritmo Fuzzy – Token Sort Ratio

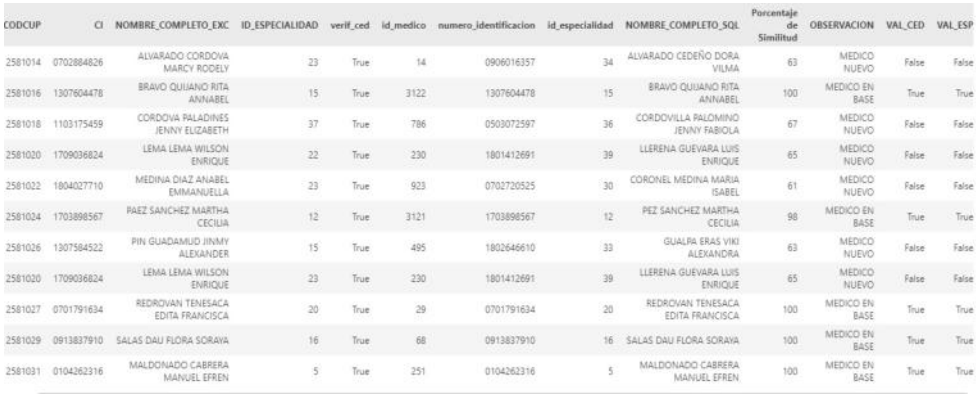
Para la ejecución del modelo fuzzywuzzy mediante algoritmo token sort ratio, el código es:

Figura 57
Código para cálculo de similitud fuzzy-token sort ratio

```
# Calcular la similitud entre los nombres y apellidos
similitud_nombre = fuzz.token_sort_ratio(nombre_excel.lower(), nombre_db.lower())
```

Obteniendo el siguiente resultado:

Figura 58
Código para visualización de resultados fuzzy-token sort ratio



CODCUP	CI	NOMBRE_COMPLETO_EXC	ID_ESPECIALIDAD	verif_ced	id_medico	numero_identificacion	id_especialidad	NOMBRE_COMPLETO_SQL	Porcentaje de Similitud	OBSERVACION	VAL_CED	VAL_ESP
2581014	0702884826	ALVARADO CORDOVA MARCY RODELY	23	True	14	0906016357	34	ALVARADO CEDEÑO DORA VILMA	63	MEDICO NUEVO	False	False
2581016	1307604478	BRAVO QUIJANO RITA ANNABEL	15	True	3122	1307604478	15	BRAVO QUIJANO RITA ANNABEL	100	MEDICO EN BASE	True	True
2581018	1103175459	CORDOVA PALADINES JENNY ELIZABETH	37	True	786	0503072597	36	CORDOVILLA PALOMINO JENNY FABIOLA	67	MEDICO NUEVO	False	False
2581020	1709036824	LEMA LEMA WILSON ENRIQUE	22	True	230	1801412691	39	LLERENA GUEVARA LUIS ENRIQUE	65	MEDICO NUEVO	False	False
2581022	1804027710	MEDINA DIAZ ANABEL EMMANUELLA	23	True	923	0702720525	30	CORONEL MEDINA MARIA ISABEL	61	MEDICO NUEVO	False	False
2581024	1703898567	PAEZ SANCHEZ MARTHA CECILIA	12	True	3121	1703898567	12	PEZ SANCHEZ MARTHA CECILIA	98	MEDICO EN BASE	True	True
2581026	1307584522	PIN GUADAMUD JINMY ALEXANDER	15	True	495	1802646610	33	GUALPA ERAS VIKI ALEXANDRA	63	MEDICO NUEVO	False	False
2581020	1709036824	LEMA LEMA WILSON ENRIQUE	23	True	230	1801412691	39	LLERENA GUEVARA LUIS ENRIQUE	65	MEDICO NUEVO	False	False
2581027	0701791634	REDROVAN TENESACA EDITA FRANCISCA	20	True	29	0701791634	20	REDROVAN TENESACA EDITA FRANCISCA	100	MEDICO EN BASE	True	True
2581029	0913837910	SALAS DAU FLORA SORAYA	16	True	68	0913837910	16	SALAS DAU FLORA SORAYA	100	MEDICO EN BASE	True	True
2581031	0104262316	MALDONADO CABRERA MANUEL EFREN	5	True	251	0104262316	5	MALDONADO CABRERA MANUEL EFREN	100	MEDICO EN BASE	True	True

4.3.5.5 Algortimo Fuzzy – Token Set Ratio

Para la ejecución del modelo fuzzywuzzy mediante algoritmo token set ratio, el código es:

Figura 59
Código para cálculo de similitud fuzzy-token set ratio

```
# Calcular la similitud entre los nombres y apellidos
similitud_nombre = fuzz.token_set_ratio(nombre_excel.lower(), nombre_db.lower())
```

Dando como resultado lo siguiente:

Figura 60
Código para visualización de resultados fuzzy-token set ratio

CODCUP	CI	NOMBRE_COMPLETO_EXC	ID_ESPECIALIDAD	verif_ced	id_medico	numero_identificacion	id_especialidad	NOMBRE_COMPLETO_SQL	Porcentaje de Similitud	OBSERVACION	VAL_CED	VAL_ESP
2581014	0702884826	ALVARADO CORDOVA MARCY RODELY	23	True	14	0906016357	34	ALVARADO CEDEÑO DORA VILMA	63	MEDICO NUEVO	False	False
2581016	1307604478	BRAVO QUIJANO RITA ANNABEL	15	True	3122	1307604478	15	BRAVO QUIJANO RITA ANNABEL	100	MEDICO EN BASE	True	True
2581018	1103175459	CORDOVA PALADINES JENNY ELIZABETH	37	True	2975	0702681693	17	HERRERA PALADINES LILIANA ELIZABETH	73	MEDICO NUEVO	False	False
2581020	1709036824	LEMA LEMA WILSON ENRIQUE	22	True	1538	0700950173	17	LOAYZA LOAYZA LUIS ENRIQUE	68	MEDICO NUEVO	False	False
2581022	1804027710	MEDINA DIAZ ANABEL EMMANUELLA	23	True	593	1801880681	35	ULLI MEDINA CARLOS EMILIANO	61	MEDICO NUEVO	False	False
2581024	1703898567	PAEZ SANCHEZ MARTHA CECILIA	12	True	3121	1703898567	12	PEZ SANCHEZ MARTHA CECILIA	98	MEDICO EN BASE	True	True
2581026	1307584522	PIN GUADAMUD JINMY ALEXANDER	15	True	495	1802646610	33	GUALPA ERAS VIKI ALEXANDRA	63	MEDICO NUEVO	False	False
2581020	1709036824	LEMA LEMA WILSON ENRIQUE	23	True	1538	0700950173	17	LOAYZA LOAYZA LUIS ENRIQUE	68	MEDICO NUEVO	False	False
2581027	0701791634	REDROVAN TENESACA EDITA FRANCISCA	20	True	29	0701791634	20	REDROVAN TENESACA EDITA FRANCISCA	100	MEDICO EN BASE	True	True
2581029	0913837910	SALAS DAU FLORA SORAYA	16	True	68	0913837910	16	SALAS DAU FLORA SORAYA	100	MEDICO EN BASE	True	True
2581031	0104262316	MALDONADO CABRERA MANUEL EFREN	5	True	251	0104262316	5	MALDONADO CABRERA MANUEL EFREN	100	MEDICO EN BASE	True	True

4.3.5.6 Modelo BERT

Otro de los modelos utilizados para el cálculo de similitud es el modelo pre-entrenado de BERT, los códigos empleados para su ejecución son:

Carga del Modelo BERT y del Tokenizador: Se carga el modelo BERT pre-entrenado y el tokenizador correspondiente, que permiten la generación de incrustaciones de palabras y la comparación de similitud semántica entre frases y oraciones.

Figura 61
Código para carga del modelo Berth y tokenizador

```
# Cargar el modelo BERT pre-entrenado y el tokenizador
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained('bert-base-uncased')
```

Iteración a través de todas las Filas: Se itera a través de todas las filas del dataframe med_nuevos2 que contiene los nombres de médicos del archivo Excel, posterior a ello se realiza el proceso de tokenizacion y word embeddings, técnica que consiste en representar palabras con vectores de números y se calcula la similitud de coseno con los nombres de médicos en la base de datos SQL.

Figura 62
Código para tokenizar todas las filas del Excel y base de médicos

```
# Iterar a través de todas las filas de MED_NUEVOS2
for _, row in med_nuevos2.iterrows():
    # Obtener el texto del nombre completo en la fila actual de MED_NUEVOS2
    texto_nuevo = row['nombre_completo_exc']

    # Tokenizar y obtener los embeddings del texto nuevo
    inputs_nuevo = tokenizer(texto_nuevo, return_tensors='pt', padding=True, truncation=True)
    with torch.no_grad():
        embeddings_nuevo = model(**inputs_nuevo).last_hidden_state.mean(dim=1)  # Promedio de embeddings de palabras

    # Tokenizar y obtener los embeddings de todos los elementos de SQL_DATA
    embeddings_sql_data = []
    for texto_sql in sql_data['nombre_completo_sql']:
        inputs_sql = tokenizer(texto_sql, return_tensors='pt', padding=True, truncation=True)
        with torch.no_grad():
            embeddings_sql = model(**inputs_sql).last_hidden_state.mean(dim=1)  # Promedio de embeddings de palabras
        embeddings_sql_data.append(embeddings_sql)

    # Calcular la similitud de coseno entre el texto nuevo y todos los elementos de SQL_DATA
    similarity_scores = []
    for embedding_sql in embeddings_sql_data:
        similarity = cosine_similarity(embeddings_nuevo, embedding_sql)
        similarity_scores.append(similarity)
```

Selección del médico con mayor similitud y Almacenamiento de Resultados: Se identifica el elemento de la base de datos SQL que tiene el mayor porcentaje de similitud con el texto nuevo, y se almacena esta información junto con otros detalles relevantes en un diccionario.

Figura 63
Código para selección de médico con mayor similitud

```
# Encontrar el índice del porcentaje de similitud más alto
max_similarity_index = similarity_percentages.index(max(similarity_percentages))

# Obtener la fila correspondiente de SQL_DATA con el porcentaje de similitud más alto
best_match_row = sql_data.iloc[max_similarity_index]
```

Creación de un DataFrame de Resultados: Se crea un DataFrame llamado result_df que contiene todos los resultados de la comparación, lo que facilita la visualización y el análisis de las coincidencias y similitudes identificadas entre los nombres de médicos.

Figura 64
Código para visualización de resultados

```
# Crear un DataFrame con los resultados
result_df = pd.DataFrame(resultados)

# Imprimir el DataFrame resultante
result_df
```

Figura 65
Visualización de resultados Modelo BERT

id_medico	numero_identificacion	id_especialidad	NOMBRE_COMPLETO_SQL	nombre_completo_exc	Similarity Percentage	VAL_CED	VAL_ESP	OBSERVACION
241	0602032930	2	ALVARADO RAMOS LUPE MARGOTH	ALVARADO CORDOVA MARCY RODELY	0.836299	False	False	MEDICO EN BASE
3122	1307604478	15	BRAVO QUIJANO RITA ANNABEL	BRAVO QUIJANO RITA ANNABEL	1.000000	True	True	MEDICO EN BASE
2975	0702681693	17	HERRERA PALADINES LILIANA ELIZABETH	CORDOVA PALADINES JENNY ELIZABETH	0.870719	False	False	MEDICO EN BASE
2944	0917627358	13	TITE AVILA LUIS ALFREDO	LEMA LEMA WILSON ENRIQUE	0.827230	False	False	MEDICO EN BASE
2814	1709279606	4	GALVEZ PEREZ MARIBEL NOEMI	MEDINA DIAZ ANABEL EMMANUELLA	0.836017	False	False	MEDICO EN BASE
3121	1703898567	12	PEZ SANCHEZ MARTHA CECILIA	PAEZ SANCHEZ MARTHA CECILIA	0.877868	True	True	MEDICO EN BASE
313	0913858072	21	GUZÑAY ENDARA MARY RUBY	PIN GUADAMUD JINMY ALEXANDER	0.790970	False	False	MEDICO EN BASE
2944	0917627358	13	TITE AVILA LUIS ALFREDO	LEMA LEMA WILSON ENRIQUE	0.827230	False	False	MEDICO EN BASE
29	0701791634	20	REDROVAN TENESACA EDITA FRANCISCA	REDROVAN TENESACA EDITA FRANCISCA	1.000000	True	True	MEDICO EN BASE
68	0913837910	16	SALAS DAU FLORA SORAYA	SALAS DAU FLORA SORAYA	1.000000	True	True	MEDICO EN BASE
251	0104262316	5	MALDONADO CABRERA MANUEL EFREN	MALDONADO CABRERA MANUEL EFREN	1.000000	True	True	MEDICO EN BASE

4.3.6 Conexión y almacenamiento de registros históricos en MongoDB

Utilizando la biblioteca Pymongo para Python se ha logrado establecer la conexión con el sistema de base de datos No SQL MongoDB, con la finalidad de almacenar los registros históricos, para ello:

Se establece la conexión con MongoDB que se ejecuta localmente "localhost", en el puerto predeterminado 27017. Posterior a ello se ha accedido a la base de datos específica para el almacenamiento llamada "laboratorio".

Figura 66
Conexión y acceso a la base Mongo DB

```
# Conexión al servidor MongoDB local
client = MongoClient("localhost", 27017)

# Accede a una base de datos
db = client.laboratorio
```

El DataFrame coincidencias_df se ha convertido en un diccionario utilizando el método to_dict con el parámetro orient='records'. Esto facilita la manipulación y la posterior inserción de los datos en MongoDB.

Figura 67
Conversión en diccionario de datos

```
# Convierte el DataFrame en un diccionario
registros_historicos = coincidencias_df.to_dict(orient='records')
```

Se ha obtenido la fecha y hora actuales utilizando el módulo datetime para marcar los registros históricos con la marca de tiempo correspondiente y se añade este dato a cada registro del diccionario.

Figura 68
Código para obtener la fecha de ejecución

```
# Obtén la fecha actual
fecha_hoy = datetime.now()

# Agregar la fecha de hoy a cada registro en el diccionario
for registro in registros_historicos:
    registro['fecha_de_ejecucion'] = fecha_hoy
```

Para poder almacenar los registros históricos se ha creado la colección "históricos" en la base de datos mencionada anteriormente.

Figura 69
Código para acceder a la colección históricos

```
# Colección en la que se almacenarán los registros históricos
historico_collection = db.historicos
```

Finalmente, se realiza la inserción de los Registros en la Colección de Historial permitiendo así la visualización, consultas y seguimiento de los cambios realizados en la base de datos.

Figura 70
Código para insertar los registros

```
# Inserta los registros en la colección de historial
historico_collection.insert_many(registros_historicos)
```

4.3.7 Carga de médicos nuevos a la base de datos SQL Server

Como último paso de este proceso, se incluye la inserción de los datos de médicos nuevos a la base de datos, para ello se ejecuta lo siguiente:

Se realiza el filtro para asegurarnos que la información insertada en la base sea únicamente de los médicos nuevos.

Figura 71
Código para filtrar información de médicos nuevos

```
# FILTRA LA INFORMACION UNICAMENTE DE LOS MEDICOS NUEVOS QUE SE CARGARAN A LA BASE
carga_base = coincidencias_df[coincidencias_df['OBSERVACION'] == 'MEDICO NUEVO']
#Seleccionamos las columnas que se van a cargar
carga_base1 = carga_base.iloc[:, :12].copy()
carga_base1.head()
```

Al igual que con los anteriores sistemas de base de datos, se establece la conexión con la base de datos SQL Server utilizando la biblioteca PYODBC y los parámetros proporcionados, que incluyen el nombre del servidor, base de datos, usuario y contraseña.

Figura 72
Código para establecer conexión con la base SQL Server

```
# Establece la cadena de conexión a tu servidor SQL Server
server = 'BFICHASIST23'
database = 'laboratorio'
username = 'sa'
password = '1719895086'
```

Se realiza una validación de la información contenida en las columnas para su inserción en la base de datos, teniendo en cuenta posibles valores nulos y el formato adecuado para campos como fechas y valores numéricos.

Figura 73
Código para validar la información a cargar

```
for index, row in carga_base1.iterrows():
    apellido_paterno = row['APELLIDO_PAT'] if not pd.isnull(row['APELLIDO_PAT']) else None
    apellido_materno = row['APELLIDO_MAT'] if not pd.isnull(row['APELLIDO_MAT']) else None
    nombres = row['NOMBRES'] if not pd.isnull(row['NOMBRES']) else None
    id_tipo_identificacion = '1'
    numero_identificacion = row['CI']

    # Comprobación de la fecha de nacimiento
    if isinstance(row['FEC_NAC'], str):
        try:
            fecha_nacimiento = datetime.strptime(row['FEC_NAC'], '%Y/%m/%d').strftime('%Y-%m-%d')
        except ValueError as e:
            print(f"Error en la fila {index}: {e}")
            fecha_nacimiento = None
    else:
        fecha_nacimiento = row['FEC_NAC'].strftime('%Y-%m-%d') if not pd.isnull(row['FEC_NAC']) else None

    email = row['EMAIL'] if not pd.isnul (module) pd ) else None
    celular = row['TEL'] if not pd.isnul  else None
    direccion = row['DIRECCION'] if not pd.isnull(row['DIRECCION']) else None

    # Comprobación de id_especialidad
    id_especialidad = row['cod_Espe']
    if not pd.isnull(id_especialidad):
        id_especialidad = int(id_especialidad) if isinstance(id_especialidad, (int, float)) else 0
    else:
        id_especialidad = None
```

Se ejecuta una consulta SQL de inserción en la tabla 'dbo.medicos' de la base de datos, con los valores proporcionados para cada columna correspondiente. Posteriormente se aprueba la ejecución de la inserción y se cierra la conexión, lo que asegura que los datos se guarden de manera efectiva y que la conexión se cierre adecuadamente después de completar la inserción.

Figura 74
Query para inserción de datos

```
    # Ejecuta la consulta
    cursor.execute("INSERT INTO dbo.medicos (apellido_paterno, apellido_materno, nombres, id_tipo_identificacion, numero_identificacion, fecha_nacimiento, email, celular, direccion, id_especialidad) VA
                   (apellido_paterno, apellido_materno, nombres, id_tipo_identificacion, numero_identificacion, fecha_nacimiento, email, celular, direccion, id_especialidad))

# Confirma la transacción y cierra la conexión
conn.commit()
conn.close()
```

CAPITULO V RESULTADOS

5.1 Impacto de negocio

Debido a la magnitud de duplicados detectados en los datos y el sector en el que opera la compañía, el valor y retorno que se proporcionará a la Compañía será:

Mejora en la calidad de los datos y toma de decisiones eficientes: la detección y eliminación de duplicados en los registros de médicos mejorará la calidad de la información, lo que conlleva a una toma de decisiones basada en datos más eficiente y efectiva.

Reducción de costos / tiempo y optimización de recursos: Al eliminar la detección manual de duplicación de registros, se pueden optimizar los recursos humanos, ya que los empleados ya no tendrán que lidiar con datos duplicados y podrán enfocarse en tareas más estratégicas.

Identificación de patrones y tendencias: Con datos de alta calidad y no duplicados, es más fácil realizar análisis de Big Data para identificar patrones, tendencias y oportunidades de negocio.

5.2 Indicadores a alcanzar

Para evaluar el rendimiento del modelo de detección de duplicados, se utilizará las siguientes métricas:

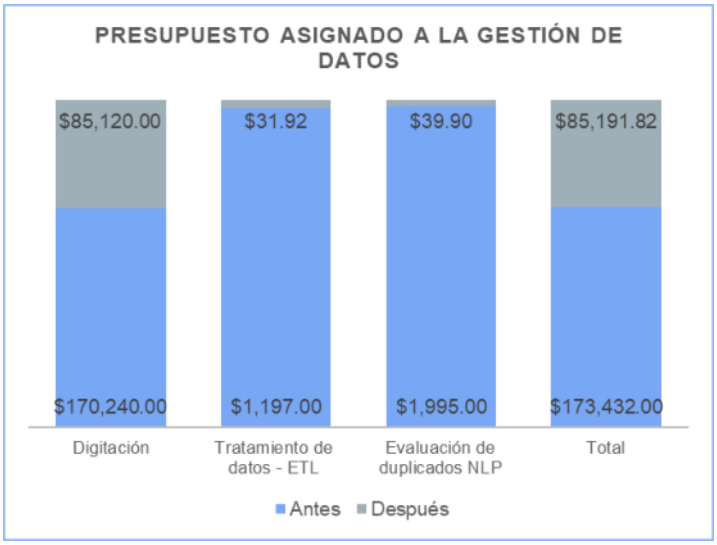
5.2.1 Porcentaje de reducción en el tiempo dedicado a la gestión de datos:

Una vez evaluadas las consecuencias de los datos duplicados en la compañía y analizando el impacto generado una vez implementado el proyecto, podemos evidenciar que existe una reducción del 73.85% en horas dedicadas a la gestión de los datos, mientras que a nivel de costos se puede ver que hay una reducción del 50.78% en los gastos implícitos en el proceso.

Tabla 15
Cálculo en horas y valores de la gestión de datos

Descripción	Antes				Después				Disminución anual	
	Población	Horas	Valor por hora	Total	Población	Horas	Valor por	Total	Horas	Valores
Tiempro dedicado a la digitación	19,200	480	$ 354.67	$ 170,240.00	19,200	240	$ 354.67	$ 85,120.00	240	$ 85,120.00
Tiempo dedicado al tratamiento de datos - ETL	19,200	180	$ 6.65	$ 1,197.00	19,200	4.80	$ 6.65	$ 31.92	175	$ 1,165.08
Tiempo dedicado a la evaluación de duplicados	19,200	300	$ 6.65	$ 1,995.00	19,200	6.00	$ 6.65	$ 39.90	294	$ 1,955.10
Total		**960**		**$ 173,432.00**		**251**		**$ 85,191.82**	**709**	**$ 88,240.18**

Figura 75
Presupuesto asignado a la gestión de datos



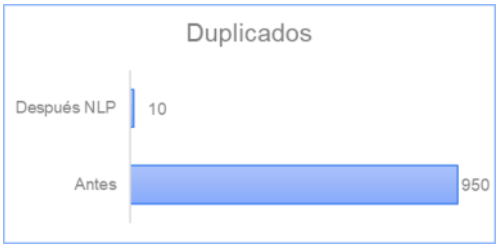
5.2.2 Porcentaje de reducción de duplicados.

Una vez implementado el proyecto en la compañía se espera tener una eficiencia del 98.95% en los datos duplicados. Logrando así reducir el impacto en los procesos posteriores a la creación de los médicos en la base de la compañía.

Tabla 16
Cálculo de reducción de médicos duplicados

Descripción	Antes	Después NLP	Diferencia	% reducción
Médicos	40,000	40,000	-	-
Duplicados	950	10	(940)	98.95%
Eficiencia	**2%**	**99.98%**		

Figura 76
Número de médicos duplicados



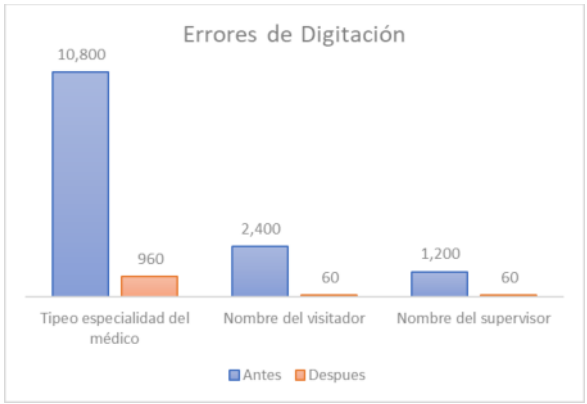
5.2.3 Porcentaje de disminución de errores en digitación

Con la implementación de un macro programa en Visual Basic, se mejorará la calidad de los datos, debido a que existirán campos con catálogos predefinidos, reduciendo así errores de digitación en un 95%, ya que los usuarios solo deben seleccionar los datos. Hemos dejado un margen de error del 5% considerando que los visitadores médicos deben seleccionar los campos y posiblemente en la selección exista errores.

Tabla 17
Cuantificación de tipo de errores en digitación

Descripción	Antes			Implementación Macros		
	Población	Errores	% error	Población	Errores	% error
Tipeo especialidad del médico	19,200	10,800	56.3%	19,200	960	5.00%
Nombre del visitador	19,200	2,400	12.5%	19,200	60	0.31%
Nombre del supervisor	19,200	1,200	6.3%	19,200	60	0.31%
Total	19,200	14,400	75.00%	19,200	1,080	5.63%

Figura 77
Tipos de Errores en digitación



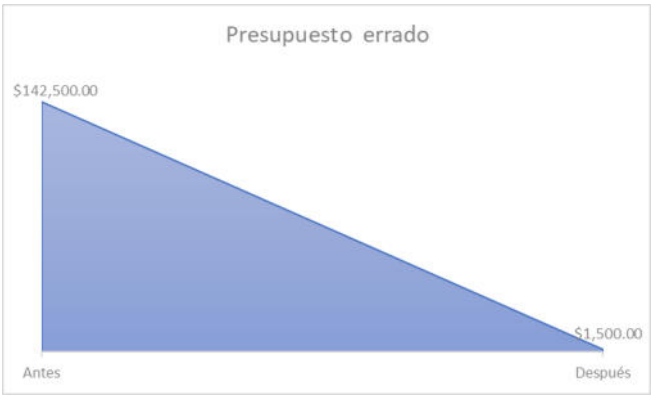
5.2.4 Presupuesto perdido asignado al médico con relación a incentivos

Anualmente se asigna un presupuesto aproximado por médico de $200, lo que corresponde a $190.000 invertidos de manera errónea, la detección temprana de duplicados ayudará a reducir esta inversión a $2.000, logrando un ahorro de $188.000.

Tabla 18
Presupuesto perdido por duplicidad en la información

Antes	Antes	Después	Ahorro
Incentivo asignado al médico promedio anual	$ 200.00	$ 200.00	200
Total médicos duplicados en la base general	950	10	940
Total presupuesto errado	$ 190,000	$ 2,000	$ 188,000

Figura 78
Variación de presupuesto de incentivos



CAPITULO VI CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- El desarrollo y la implementación de un modelo de procesamiento de lenguaje natural (NLP) para detectar automáticamente datos duplicados en conjuntos de datos de texto representa un avance importante para la empresa en la optimización de la gestión de datos y la mejora de la calidad de la información. El proyecto demostró su capacidad para abordar desafíos clave asociados con la identificación manual de duplicados al proporcionar una solución automatizada y eficiente para mejorar la precisión y la integridad de los conjuntos de datos.
- La implementación de este modelo no solo demostró su potencial para aumentar la eficiencia operativa y reducir los riesgos y costos asociados con datos duplicados, sino que también mejoró significativamente la experiencia del usuario al garantizar resultados más precisos y relevantes. Además, el cumplimiento normativo se mejora mediante una mayor integridad y confidencialidad de los datos.
- Este proyecto se alinea perfectamente con la iniciativa de integración de nuevas tecnologías propuesta por la empresa, demostrando el compromiso de la organización con la innovación y la transformación digital. La adopción de este modelo NLP demuestra la voluntad de la empresa de mantenerse a la vanguardia de la evolución tecnológica, garantizando una gestión de datos más efectiva y una toma de decisiones más informada en un entorno empresarial en constante cambio.
- La implementación exitosa de este modelo NLP para la detección de datos duplicados no solo impulsará la eficiencia y la calidad en la gestión de datos, sino que también reforzará la posición de la empresa como líder en la aplicación de tecnologías avanzadas para mejorar los procesos empresariales y la experiencia del cliente.

6.2 Recomendaciones

- Fomentar una comunicación abierta y regular entre el equipo de desarrollo y el cliente para comprender completamente las necesidades y requisitos específicos. Asegurando alineación constante y la resolución oportuna de problemas que puedan surgir durante el desarrollo y la implementación del proyecto. Posterior a la implementación se recomienda realizar evaluaciones periódicas de las necesidades y expectativas del cliente, lo que permitirá realizar ajustes y mejoras para garantizar que el modelo NLP aborde de manera efectiva las demandas cambiantes del entorno empresarial y de los usuarios finales.
- Dado la eficiencia demostrada del proyecto en este proceso especifico, se recomienda integrar estratégicamente modelos de NLP en los distintos sistemas y áreas de la compañía. Esto garantizará una notable mejora en la eficiencia operativa, así como también consolidará la calidad de la información y reforzará la integridad de los datos en toda la organización.
- Establecer políticas y procedimientos claros para garantizar el uso y manejo adecuado de los datos personales recopilados durante la implementación del modelo NLP. Dando cumplimiento a las normativas vigentes de protección de datos y fortalece la confianza del cliente en el manejo ético y responsable de la información confidencial.
- Realizar auditorías periódicas de seguridad de datos para garantizar que los programas y códigos implementados cumplan con los estándares de seguridad y privacidad requeridos por la normativa vigente. Esto ayudará a identificar y abordar posibles brechas de seguridad de manera proactiva, mitigando así cualquier riesgo potencial de incumplimiento, protegiendo la reputación y la confianza de la empresa y sus clientes.

REFERENCIAS BIBLIOGRÁFICAS

Amit Phaltankar, J. A. (2020). *MongoDB Fundamentals.* Birmingham, UK : Packt Publishing Ltd. .

Chandra, P. (20 de Mayo de 2023). *Medium*. Obtenido de https://medium.com/@chandu.bathula16/understanding-fuzzy-string-matching-exploring-fuzz-ratio-fuzz-partial-ratio-token-set-ratio-and-d6892430f53c

Chowdhary, K. R. (2020). *Fundamentals of Artificial Intelligence.* New Delhi, India: Springer.

Devlin, J. C.-W. (2008). *Bert: Pre-training.*

Dirección_Nacional_de_Registros_Públicos. (2021). Ley de Protección de Datos Personales. Obtenido de https://www.finanzaspopulares.gob.ec/.

Dutta, M. (19 de Septiembre de 2023). *Analytics Vidhya*. Obtenido de https://www.analyticsvidhya.com/blog/2021/07/fuzzy-string-matching-a-hands-on-guide/#h-partial-ratio-using-fuzzywuzzy

Forta, B. (2004). *Teach Yourself SQL.* Indianapolis, USA: SAMS.

Fred R, D. (2011). *Strategic Management Concepts and Cases.*

Gallego, M. T. (2019). *«Metodología SCRUM.* Obtenido de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memor

GitHub. (2023). Obtenido de https://github.com/seatgeek/fuzzywuzzy.

Github. (2023). *https://github.com/avian2/unidecode*. Obtenido de https://github.com.

Guido, A. C. (2017). *Introduction to Machine Learning.* EEUU: O'Reilly Media, Inc.

Hemachandran K., R. R. (2023). *Artificial Intelligence for Business.* New York, EEUU: Routledge.

IBM. (2023). Obtenido de ¿Qué es una base de datos relacional?: https://www.ibm.com/mx-es/topics/relational-databases

IBM. (2023). *¿Qué es el procesamiento del lenguaje natural (NLP)?* Obtenido de IBM: https://www.ibm.com/es-es/topics/natural-language-processing

IBM. (15 de 11 de 2023). *https://www.ibm.com/es-es/topics/natural-language-processing*. Obtenido de IBM Natural-language-processing: https://www.ibm.com/es-es/topics/natural-language-processing

IBMCloudEducation. (2019). *IBM Cloud Learn Hub*. Obtenido de https://www.ibm.com/cloud/learn/nosql-databases.

IBMEducation. (2019). *IBM Cloud Learn Hub*. Obtenido de https://www.ibm.com/cloud/learn/relational-databases

ISO-ORG. (2023). *https://www.iso.org/standard/35736.html*. Obtenido de https://www.iso.org/standard/35736.html.

Johnson Gerry, S. K. (2005). *Exploring Corporate Strategy.*

Jurafsky, D. &. (2020). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall.

Kaufmann, A. M. (2019). *SQL & NoSQL Databases.* Alemania: Springer Vieweg.

Kaur, A. (2019). *GeeksforGeeks*. Obtenido de https://www.geeksforgeeks.org/need-for-dbms/.

M. Radivojevic, D. S. (2019). *Mastering SQL Server 2017.* Birmingham, U.K.: Pack Publishing Ltd.

McKinney, W. (2013). *Python for Data Analysis.*

Microsoft. (2023). *https://code.visualstudio.com/*. Obtenido de https://code.visualstudio.com/.

MicrosoftDocumentation. (2018). *Microsoft Documentation.* Obtenido de https://docs.microsoft.com/en-us/visualstudio/ide/using-intellisense?view=vs-2019.

MongoDB. (2023). *https://www.mongodb.com*. Obtenido de https://www.mongodb.com.

Pang, B. L. (2008). *Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval.*

Parlamento_Europeo. (2016). *https://www.boe.es/doue/2016/119/L00001-00088.pdf.* Obtenido de https://commission.europa.eu/index_es.

Pykes, K. (Marzo de 2023). *Datacamp*. Obtenido de https://www.datacamp.com/tutorial/fuzzy-string-python

PyMongo. (2023). *https://pymongo.readthedocs.io/en/stable/*. Obtenido de https://pymongo.

Scikit-learn_machine learning in Python-scikit-learn 1.0.2 documentation, s. (2023). *https://scikit-learn.org/stable/* . Obtenido de https://scikit-learn.org/stable/ .

Sullivan, D. (2015). *NoSQL for Mere Mortals.* Michigan, US: Pearson Education, Inc.

Vasilev, I. (2019). *Python Machine Learning.*

VisualStudioCode. (2020). *Visual Studio Code.* Obtenido de https://code.visualstudio.com/docs/editor/whyvscode

Wolf, T. D. (2019). *HuggingFace's Transformers: State-of-the-art Natural Language Processing.*

APÉNDICES

7.1 Cronograma y planificación

Tabla 19
Cronograma y planificación

	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16
Diseño de un modelo de detección de datos duplicados mediante NLP																	
Fase 1: Planificación (Semana 1-4)																	
Definición de objetivos y alcance del proyecto.																	
Identificación de los equipos y recursos necesarios.																	
Revisión de la literatura existente sobre detección de datos duplicados en PLN.																	
Establecimiento de métricas de evaluación de rendimiento.																	
Fase 2: Recopilación de Datos (Semana 5-8)																	
Identificación de las fuentes de datos relevantes en el laboratorio farmacéutico.																	
Definir el flujo del proceso.																	
Elección del motor de base datos a utilizar.																	
Creación de una base de datos de entrenamiento y prueba.																	
Conexión de las bases de datos.																	
Elaboración de la documentación del avance realizado.																	
Fase 3: Desarrollo del Modelo (Semana 9-12)																	
Selección de algoritmos de NLP adecuados para la detección de duplicados.																	
Desarrollo del modelo.																	
Evaluación del rendimiento del modelo en datos de prueba.																	
Ajustes finales del modelo																	
Incorporación de los avances realizados a la documentación.																	
Fase 4: Presentación del proyecto(Semana 13-16)																	
Revisión y ajuste del documento escrito.																	
Elaboración de la presentación y exposición del proyecto.																	

7.2 Planificación de recursos

Para el desarrollo de este proyecto la planificación de recursos es la siguiente por horas y perfil:

Tabla 20
Planificación de recursos

Diseño de un modelo de detección de datos duplicados mediante NLP	TOTAL HORAS	INGENIERO DE DATOS	ARQUITECTO DE DATOS	CIENTIFICO DE DATOS	JEFE DE PROYECTO
Fase 1: Planificación	60,00	0,00	0,00	0,00	60,00
Definición de objetivos y alcance del proyecto.	10,00				10,00
Identificación de los equipos y recursos necesarios.	10,00				10,00
Revisión de la literatura existente sobre detección de datos duplicados en PLN.	30,00				30,00
Establecimiento de métricas de evaluación de rendimiento.	10,00				10,00
Fase 2: Recopilación de Datos	120,00	20,00	50,00	20,00	30,00
Identificación de las fuentes de datos relevantes en el laboratorio farmacéutico.	30,00			20,00	10,00
Definir el flujo del proceso.	20,00		20,00		
Elección del motor de base datos a utilizar.	10,00		10,00		
Creación de una base de datos de entrenamiento y prueba.	20,00		20,00		
Conexión de las bases de datos.	20,00	20,00			
Elaboración de la documentación del avance realizado.	20,00				20,00
Fase 3: Desarrollo del Modelo	120,00	25,00	25,00	90,00	30,00
Selección de algoritmos de NLP adecuados para la detección de duplicados.	40,00	10,00	10,00	20,00	
Evaluación del rendimiento del modelo en datos de prueba.	20,00			20,00	
Ajustes finales del modelo	30,00	5,00	5,00	20,00	
Incorporación de los avances realizados a la documentación.	20,00				20,00
Realización de pruebas de factibilidad del proceso	10,00				10,00
Fase 4: Presentación del proyecto	50,00	0,00	0,00	0,00	50,00
Revisión y ajuste del documento escrito.	30,00				30,00
Elaboración de la presentación y exposición del proyecto.	20,00				20,00

7.3 Documentación técnica del proyecto

La documentación técnica del presente proyecto se encuentra disponible en el siguiente repositorio público de GitHub:

https://github.com/jacqueline-quispe/DETECCION-DE-DUPLICADOS-LABORATORIO-FARMACEUTICO/blob/main/README.md

Manual técnico:

https://github.com/jacqueline-quispe/DETECCION-DE-DUPLICADOS-LABORATORIO-FARMACEUTICO/blob/main/MANUAL%20T%C3%89CNICO.docx

Manual de usuario:

https://github.com/jacqueline-quispe/DETECCION-DE-DUPLICADOS-LABORATORIO-FARMACEUTICO/blob/main/MANUAL%20DE%20USUARIO.docx

GLOSARIO

Abreviatura	Descripción
ACID	Atomicity, Consistency, Isolation and Durability, por su nombre en inglés, Atomicidad, Coherencia, integridad y Durabilidad
AD	Arquitecto de datos
APEC	Asia-Pacific Economic Cooperation, por su nombre en inglés, la asociación de Asia y el Pacífico
APF	Adaptive Project Framework, por su nombre en inglés, Marco de Proyecto Adaptativo
API	Application Programming Interface, por su nombre en inglés, Interfaz de Programación de Aplicaciones
AUC	Area under the curve, por su nombre en inglés, Area bajo la curva
BERT	Bidirectional Encoder Representations from Transformers, por su nombre en inglés, Representaciones de Codificador Bidireccional de Transformadores
CBPR	Cross-Border Privacy Rules, por su nombre en inglés, Reglas de Privacidad Transfronterizas
CCPA	California Consumer Privacy Act, por su nombre en inglés, Ley de Privacidad del Consumidor de California
CD	Científico de datos
CRM	Customer Relationship Management, por su nombre en inglés, Gestión de Relaciones con el Cliente
CSL	Cyber Security Law, por su nombre en inglés, Ley de Seguridad Cibernética
DBMS	Database Management System, por su nombre en inglés, Sistema de Gestión de Bases de Datos
DINARDAP	Dirección Nacional de Registro de Datos Públicos
DPA	Data Protection Act, por su nombre en inglés, Ley de Protección de Datos del Reino Unido
DPO	Delegado de Protección de Datos
DSDM	Dynamic Systems Development Method, por su nombre en inglés, Método de Desarrollo de Sistemas dinámicos
E/R	Entidad-Relación
ERP	Enterprise Resource Planning, por su nombre en inglés, Planificación de Recursos Empresariales
ETL	Extract Transform and Load, por su nombre en inglés, Extraer, Transformar y Cargar
FDD	Feature Driven Development, por su nombre en inglés, Desarrollo Basado en Funciones
GPS	Global Positioning System, por su nombre en inglés, Sistema de Posicionamiento Global

Abreviatura	Descripción
HIPAA	Health Insurance Portability and Accountability, por su nombre en inglés, Ley de Transferencia y Responsabilidad de Seguro Médico
HTTP	Hypertext Transfer Protocol, por su nombre en inglés, Protocolo de Transferencia de Hipertexto
IA	Inteligencia Artificial
ICDPPC	International Conference of Data Protection and Privacy Commissioners, por su nombre en inglés, Conferencia Internacional de Comisionados de Protección de Datos y Privacidad
ID	Identification, por su nombre en inglés, Identificación
ID	Ingeniero de datos
IDE	Integrated Development Environment, por su nombre en inglés, Entorno de Desarrollo Integrado
IoT	Internet Of Things, por su nombre en inglés, Internet de las cosas
JP	Jefe de proyecto
JSON	JavaScript Object Notation, por su nombre en inglés, Notación de objetos JavaScript
LFPDPPP	Ley Federal de Protección de Datos Personales en Posesión de los Particulares
LGPD	Lei Geral de Proteção de Dados, por su nombre en portugués, Ley General de Protección de Datos
LOPD	Ley Orgánica de Protección de Datos
LOPDP	Ley Orgánica de Protección de Datos Personales
LPVP	Ley de Protección de la Vida Privada
ML	Machine Learning, por su nombre en inglés, aprendizaje automático
NEM	Reconocimiento de Entidad Denominada
NLP	Natural Language Processing, por su nombre en inglés, Procesamiento del Lenguaje Natural
NORMA ISO/IEC	Organización Internacional de Normalización / Comisión Electrotécnica Internacional
NoSQL	No Structured Query Language, por su nombre en inglés, Bases de Datos no Relacionales
ODBC	Open Database Connectivity, por su nombre en inglés, Conectividad de Base de Datos Abierta
PDPA	Personal Data Protection Commission, por su nombre en inglés, Comisión de Protección de Datos Personales
PESTEL	Political, Economic, Social, Technological, Legal, and Environmental, por su nombre en inglés, Político, Económico, Social, Tecnológico, Legal y Ambiental
PIPA	Personal Information Protection Act, por su nombre en inglés, Protección de Información Personal
RGPD	Reglamento General de Protección de Datos Personales

Abreviatura	Descripción
SAFe	Scaled Agile Framework, por su nombre en inglés, Marco ágil escalado
SENESCYT	Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación
SQL	Structured Query Language, por su nombre en inglés, Bases de Datos Relacionales
SSMS	Server Management Studio, por su nombre en inglés, Estudio de Gestión de Servidores
TI	Tecnología de la Información
TTD	Tecnología y Transformación Digital
UE	Unión Europea
URL	Uniform Resource Locator, por su nombre en inglés, Localizador de Recursos Uniforme
XML	eXtensible Markup Language, por su nombre en inglés, lenguaje de marcado extensible
XP	eXtreme Programming, por su nombre en inglés, Programación Extrema