UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TÍTULO DEL TRABAJO DE TITULACIÓN

DESARROLLO DE UN AGENTE PARA LA DETECCIÓN DE SPAM EN EL SERVICIO DE CORREO ELECTRÓNICO ZIMBRA APLICANDO TÉCNICA DE MACHINE LEARNING DE CLASIFICACIÓN DE TEXTO PARA UN GAD MUNICIPAL.

AUTOR

SORIA MÉNDEZ BRYAN ANDRÉS

PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Lsi. DANIEL QUIRUMBAY YAGUAL, MSIA

Santa Elena, Ecuador

Año 2023

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y

TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino. Mgt.
DIRECTOR DE LA CARRERA

Lsi. Daniel Quirumbay Yagual, Mgt.
TUTOR

Ing. Iván Coronel Suárez. Mgt.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez. Mgt.
DOCENTE GUÍA UIC

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Bryan Andrés Soria Méndez, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 02 días del mes de agosto del año 2023

TUTOR

Firmado electrónicamente por:
DANIEL IVAN QUIRUMBAY YAGUAL

Lsi. Daniel Quirumbay, Msia

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Bryan Andrés Soria Méndez**

DECLARO QUE:

El trabajo de Titulación, "Desarrollo de un agente para la detección de spam en el servicio de correo electrónico Zimbra aplicando técnica de machine learning de clasificación de texto para un GAD municipal" previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 02 días del mes de agosto del año 2023

EL AUTOR

BryanSM

Bryan Andrés Soria Méndez

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado **"Desarrollo de un Agente para la detección de spam en el servicio de correo electrónico Zimbra aplicando técnica de machine learning de clasificación de texto para un gad municipal",** presentado por el estudiante, Bryan Andrés Soria Méndez fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 6%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.


CERTIFICADO DE ANÁLISIS
magister

Soria_Mendez_Bryan_Titulacion2023

6% Similitudes

4% Texto entre comillas
1% similitudes entre comillas

2% Idioma no reconocido

Nombre del documento: Soria_Mendez_Bryan_Titulacion2023.docx
ID del documento: 1c461e8ee2c1bea9cbf90386fe93194c41a19350
Tamaño del documento original: 7,92 MB

Depositante: DANIEL IVAN QUIRUMBAY YAGUAL
Fecha de depósito: 2/8/2023
Tipo de carga: interface
fecha de fin de análisis: 2/8/2023

Número de palabras: 30.988
Número de caracteres: 207.432

TUTOR

Firmado electrónicamente por:
DANIEL IVAN QUIRUMBAY YAGUAL

Lsi. Daniel Quirumbay, Msia

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Bryan Andrés Soria Méndez**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 02 días del mes de agosto del año 2023

EL AUTOR

BryanSM

Bryan Andrés Soria Méndez

AGRADECIMIENTO

A mis padres por brindarme su incondicional apoyo durante toda esta etapa de mi carrera y mi vida, por haber estado pendiente de mi en todo momento.

A mis hermanos, quiero expresarles mi más profundo agradecimiento por aportar un granito de arena durante este proceso.

Agradecer a los docentes que han brindado su conocimiento y las herramientas necesarias durante el estudio de esta carrera.

Bryan Andrés, Soria Méndez

DEDICATORIA

Este proyecto va dedicado a mis padres y toda mi familia en general, por brindarme su apoyo sin dudarlo en momentos que los necesitaba.

También a mis compañeros que me acompañaron durante toda la etapa de estudio y los momentos de incertidumbre.

A los docentes que han sido parte fundamental de mi etapa estudiantil y en el desarrollo de mi tema de titulación

Bryan Andrés, Soria Méndez

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

Resumen

El GAD municipal tiene como objetivo principal brindar servicios a toda la comunidad a nivel municipal, actualmente cuenta con más de 40 departamentos dentro de la organización y todos ellos son cubiertas por los servicios que brinda el área de tecnología, una de ellas es el servicio de correo electrónico Zimbra. Actualmente el GAD cuenta con correos electrónicos categorizados como spam y carecen de un control para identificar y mitigar los correos spam dentro del servicio, por lo que se propone el desarrollo de un agente que detecte los correos electrónicos categorizados como spam mediante el uso de un algoritmo de aprendizaje automático que ayude a remover los correos electrónicos detectados como no deseados a la carpeta de spam del usuario, alertando al mismo sobre la detección y brindándole la opción de reportarlo. De esta manera, el administrador puede tomar medidas evitar que los correos spam sean transmitidos hacia el servicio de Zimbra del GAD.

Palabras claves: Aprendizaje automático, Zimbra, Python, Spam.

Abstract

The main objective of the municipal government is to provide services to the entire community at the municipal level, currently has more than 40 departments within the organization and all of them are covered by the services provided by the technology area, one of them is the Zimbra email service. Currently the GAD has emails categorized as spam and lacks a control to identify and mitigate spam emails within the service, so it is proposed to develop an agent that detects emails categorized as spam by using a machine learning algorithm that helps to remove emails detected as unwanted to the user's spam folder, alerting the user about the detection and providing the option to report it. In this way, the administrator can take action to prevent spam emails from being forwarded to the GAD Zimbra service.

Keywords: Machine learning, Zimbra, Python, Spam.

Introducción

El presente proyecto tiene como objetivo desarrollar un agente que detecte en tiempo real correos electrónicos spam en el servicio de Zimbra aplicando diversas metodologías y técnicas tales como la KDD (descubrimiento conocimiento útil) que permitirá obtener un conjunto de datos listo para el entrenamiento del algoritmo de aprendizaje automático seleccionado, seguido de la OMSTD que permitirá seguir estándares para el desarrollo del agente y por finalizar la metodología CSF de la institución NIST, que permitirá justificar las ventajas que tendrá la implementación del agente.

En el capítulo 1 se explicará información relacionada a la empresa y la problemática que cuentan con relación al spam en el servicio de Zimbra, también las metodologías más detalladas y el alcance que se tendrá en cada una de ellas, describiendo que se realizará en cada fase.

En el capítulo 2 se detallará el proceso para poder desarrollar el agente detector de spam siguiendo y abordando cada una de las fases de las metodologías aplicadas. A lo largo de este capítulo se explicará los pasos seguidos en cada una de las etapas.

CAPÍTULO 1.

1. Fundamentación

1.1.Antecedentes

Los ciberdelincuentes siguen usando el correo electrónico como medio principal para la distribución de software malicioso, así como también propagar código de minería de bitcoin y realizar phishing [1]. Según reportes de Kapersky Lab sobre el spam y phishing en el 2016, las alertas se activaron 239.979.660 veces en los ordenadores de los usuarios que hacen uso de sus servicios. Según estadísticas de Kapersky en el año 2021, el 45,56% de los correos electrónicos eran spam y se evitó que los usuarios entren a 314.954 enlaces que en su contenido incluía suplantación de identidad [2].

El GAD municipal público, tiene como objetivo atender a la ciudadanía en general para ofrecer servicios a su comunidad, cuenta con 47 departamentos y 711 empleados [3], donde se incluye el departamento de sistemas que ofrece distintos servicios a todo el edificio, tales como asistencia técnica, desarrollo, vigilancia de cámaras de seguridad, correo electrónico, entre otros.

La afluencia de usuarios en el GAD es de 100 personas (promedio por día), considerando que las personas acuden a diferentes departamentos de la empresa conforme del tipo de problema que tengan o el trámite que desean realizar [4].

Los resultados obtenidos en el Anexo 2, nos explica que algunos de los correos electrónicos que circulan dentro de la empresa son categorizados como spam o de dudosa procedencia, donde los usuarios pueden ser víctimas y exponer información confidencial personal o de la empresa. De igual manera, todas las peticiones o solicitudes internas de la empresa son realizadas a través de correo electrónico, siendo una brecha de seguridad donde se pueden enviar y recibir información que contenga spam.

En el Anexo 3, se menciona que el GAD cuenta con firewall que filtre información entrante a la red. Sin embargo, se observa que, a pesar de aplicar dicha medida de seguridad, no se filtran los mensajes como spam y no se alertan automáticamente a

los administradores. Además, no se cuenta con un sistema que alerte a los usuarios de que le ha llegado un correo electrónico como spam, solo se realiza este proceso mediante los reportes diarios que el servicio de Zimbra le entrega al administrador de todos los usuarios que han enviados correos en un día.

En el Anexo 4, Anexo 5 y Anexo 6, se puede identificar ejemplos de correos electrónicos que contienen spam, donde en ambos casos los dominós remitentes del mensaje comparten siglas relacionadas a entidades públicas o gubernamentales, como “gob”, “ec” y “pe”, estos dos últimos haciendo referencia a los códigos de los países ecuador y Perú respectivamente. En el Anexo 5 se ve que el remitente y receptor del correo son iguales

El trabajo realizado en la Universidad Politécnica de Cataluña, con el tema “Detección de Spam mediante aprendizaje automático basado en el análisis de texto” [5], se realiza una investigación sobre el uso de los algoritmos de aprendizaje automático para el análisis de texto de los correos electrónicos, donde desarrollaron una herramienta en fase de prueba, donde permite al usuario poder autenticarse con su cuenta de correo electrónico y detectar si un correo es phishing, donde aplicaron diferentes algoritmos de aprendizaje automático. Sin embargo, se requiere que durante su etapa de desarrollo cada usuario que quiera hacer uso de ella deberá de instalarlo en su computador personal, por lo que el enfoque para plasmarlo en un entorno de producción puede causar problemas de eficiencia y escalabilidad.

En la universidad de los Andes se desarrolló el tema “Machine Learning y Seguridad: Detección de Correos Falsos y Detección de Intrusos”[6], empleando los algoritmos SVM (Máquinas de vectores de soporte) y Random Forest previamente seleccionados, haciendo uso de las bases de datos “Nazario Phishing Corpus” y “Enron email”, sin embargo, no realizan las pruebas en tiempo real y lo hacen en un entorno controlado.

En la escuela politécnica nacional de quito, se propuse el tema “Análisis y diseño de un modelo predictivo para detección de phishing basado en url y corpus del correo electrónico” [7], empleando análisis de las URL que estén incluidas en el

cuerpo del correo electrónico, sin embargo, usan datasets que no son actualizadas y no se emplea en tiempo real.

En los trabajos citados previamente, todos hacen un análisis previo de los algoritmos de aprendizaje automático a usar en cada una de sus pruebas, pero cada una de estas son empleadas en entornos controlados o de pruebas, tal que no se pueda generar gran cantidad de información como se lo podría generar con una empresa, además, solo emplearlas para detectar correos spam a una sola cuenta de usuario, por lo que puede resultar complicado realizar un escalamiento.

Detallando los problemas, se desarrollará un agente para la detección en tiempo real y prevención de correos electrónicos con contenido spam dentro del servicio Zimbra, permitiendo una rápida acción frente a estos casos y evitar posibles intentos de ingeniería social hacia los trabajadores del GAD, añadiendo una capa extra de seguridad para sobreproteger los activos de información y reducir posibles riesgos de incidentes de seguridad.

1.2.Descripción del proyecto

La institución municipal posee un servicio de correo electrónico que permite el envío y recepción de correos electrónicos de dominios internos y externos, de tal forma que los usuarios pueden comunicarse sin importar el origen o destinos de los mensajes, generando grandes actividades. Principalmente los usuarios hacen uso de este servicio para realizar solicitudes para el pedido de recursos, asignaciones de reuniones, entre otras actividades para cumplir con los requerimientos de la empresa.

El limitado control que existe para identificar si un correo electrónico incluye contenido spam, información engañosa, archivos con código malicioso o que el usuario no pueda identificar este tipo de engaños y acceda a realizar este tipo de acciones, pone en riesgo la información confidencial de la empresa.

El proyecto consiste en el desarrollo de un agente para la detección de spam en correos electrónicos en el servicio de Zimbra, con el fin de remover el correo electrónico detectado como no deseado hacia a la bandeja de spam por el agente, posteriormente se enviará una alerta al usuario y este podrá reportar el correo

electrónico como spam, donde el administrador puede con el previo consentimiento del usuario ver el mensaje detectado como spam, para posteriormente poder aplicar decisiones, de esta manera se añade una capa de seguridad a este servicio.

Para el correcto desarrollo del agente de detección de spam en correos electrónicos, se implementarán tres metodologías: KDD (Descubrimiento de conocimiento en bases de datos), OMSTD (Metodología abierta para desarrolladores de herramientas de seguridad) y CSF (Marco de ciberseguridad).

Metodología KDD

La detección de correos electrónicos como spam será por medio de un análisis predictivo, donde la metodología KDD proporciona un marco estructurado y sistemático para descubrir conocimiento útil a partir de grandes cantidades de volúmenes de datos, combinando descubrimiento y análisis [8]. Las fases de la metodología son:

- Selección.
- Preprocesamiento/limpieza.
- Transformación/reducción.
- Minería de datos.
- Interpretación/evaluación.

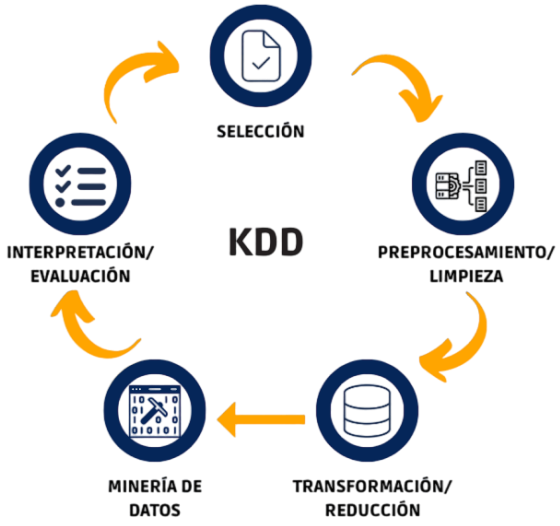


Fig. 1. Metodología de descubrimiento conocimiento útil [8].

Selección

- Selección de varias fuentes confiables que contengan conjuntos de datos que incluyan correos electrónicos categorizados como spam y normal.
- Selección de conjunto de datos apropiados para el análisis y detección de correos electrónicos como spam.

Preprocesamiento/limpieza

- Proceso de limpieza y preparación de los datos obtenidos en la fase anterior.
- Traducción al idioma español de los conjuntos de datos seleccionados.
- Identificación y eliminación de atributos irrelevantes para la detección de spam.
- Aplicar técnica de tokenización para la división de textos del conjunto de datos en unidades más pequeñas.

Transformación/reducción

- Combinación de los conjuntos de datos previamente obtenidos.

Minería de datos

- Selección de algoritmos de aprendizaje automático.
- Entrenamiento de los algoritmos seleccionados con el conjunto de datos preprocesados.
- Se mostrará tabla estadística con los resultados obtenidos por cada algoritmo

Interpretación/evaluación

- Evaluación de los datos obtenidos en la fase de minería de datos.
- Comparación de los resultados obtenidos de cada algoritmo en términos de desempeño y eficacia.
- Selección del algoritmo con mejores resultados.

Metodología OMSTD

Es una metodología y conjunto de buenas prácticas para lograr el desarrollo de herramientas de seguridad bien construidas, basándose principalmente en herramientas de hacking escritas en Python, aunque no se limita especialmente a este lenguaje [9]. Sus fases son las siguientes:

- Organización y estructura
- Entrada y salida de información
- Redistribución
- Despliegue

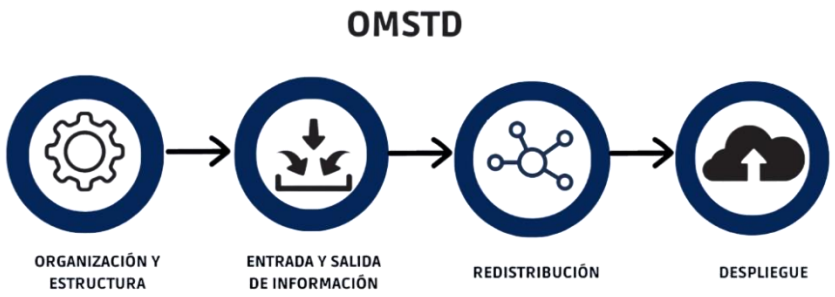


Fig. 2. Metodología OMSTD para el desarrollo del agente [9].

Organización y estructura

- Comparativa y selección entre lenguajes de programación.
- Comparativa y selección entre marcos de trabajos del lenguaje seleccionado.

Entrada y salida de información

- Definir los tipos de entrada y salida de información.

Redistribución

- Se realizará la programación de cada una de las partes que engloben la arquitectura del agente de detección de spam.
- Implementar e integrar cada una de las partes que conforman la arquitectura del agente que detecte spam.

Despliegue

- Configuración del entorno de despliegue para la implementación de la herramienta.
- Despliegue de la herramienta en el entorno de prueba.
- Despliegue de la herramienta en el servicio de correo electrónico Zimbra del GAD.

Marco de ciberseguridad

El marco de ciberseguridad CSF cuenta con tres componentes, que son el núcleo, los niveles y los perfiles. En el núcleo del marco se aplican cinco funciones esenciales para el cumplimiento de este y son: identificar, proteger, detectar, responder y recuperar [10]. Los Niveles de Implementación del Marco representan el grado en el que la organización ha implementado y madurado sus capacidades de ciberseguridad, mientras que los Perfiles del Marco permiten adaptar el enfoque de ciberseguridad a las necesidades específicas de la organización [10].

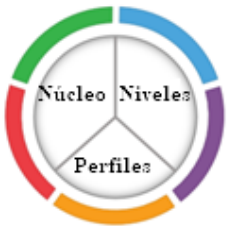


Fig. 3. Componentes del marco de ciberseguridad CSF [10].

La metodología CSF del NIST (Instituto Nacional de Normas y Tecnología) es una guía para la gestión de ciberseguridad, se compone de un núcleo y se aplican cinco funciones esenciales para el cumplimiento de los objetivos de seguridad y son: identificar, proteger, detectar, responder y recuperar [10].

Fig. 4. Componente núcleo del marco de trabajo CSF [10].

Núcleo

Fase 1 Identificar

Entendimiento de la organización.

Se llevará a cabo una recopilación de datos con el fin de comprender mejor la organización y el servicio de correo electrónico que se utiliza actualmente. Esta recopilación de datos incluirá los siguientes aspectos:

- Las estrategias de ciberseguridad que se aplican en el servidor de correo electrónico.
- La cantidad de usuarios que poseen cuenta una de correo electrónico.
- Los activos que posee el servidor de correo electrónico.

Fase 2 Proteger

Se llevará a cabo recolección de información sobre las técnicas que se hacen uso para proteger los activos. La recopilación de datos incluirá:

- Identificación de políticas de gestión de identidad, autenticación y control de acceso que se emplean en el servicio de correo electrónico
- Identificación de las técnicas que usan para proteger los datos.

Fase 3 Detectar

Se llevará a cabo recolección de información acerca de los métodos actuales que usan para poder detectar a un correo electrónico como spam. La recopilación de datos incluirá:

- Identificación de herramientas que aplican para poder detectar un correo electrónico como spam

Fase 4 Responder

La recolección de información durante esta fase será relacionada con:

- Identificar si existen medios de alertas contra correos spam.
- Identificar si se analizan las alertas.

- Identificar si se emplean medidas para evitar la propagación de spam.

Fase 5 Recuperar

Durante esta fase se identificarán las siguientes informaciones:

- Identificación de planes que existen para poder recuperar los correos electrónicos en caso de incidentes

Niveles de implementación

En cada una de las fases mencionadas previamente, se procederán a establecer lo siguiente:

- Identificación del nivel de implementación de cada subcategoría relacionadas al servicio de correo electrónico

Perfiles

En cada una de las categorías identificadas previamente, se procederá a:

- Identificación de los perfiles actuales de cada subcategoría por cada función que proporciona la metodología CSF y adaptarla al servicio de correo electrónico.
- Establecer un perfil objetivo de cada subcategoría del servicio de correo electrónico

1.2.1. Herramientas

Visual Studio Code:

Es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET) [11].

Python:

Es un lenguaje de programación interpretada, orientada a objetos y de alto nivel con semántica dinámica, considerado lenguaje de alto nivel por su sintaxis clara y legible que facilita la lectura y escritura del código [12].

Pandas:

Es una herramienta de análisis y manipulación de datos de código abierto diseñado específicamente para el lenguaje de programación Python, permitiendo trabajar con datos de una manera rápida y eficiente [13]**.**

Flask:

Es un microframework de Python, enfocado en proporcionar funcionalidades esenciales para el desarrollo de páginas webs sin agregar funcionalidades innecesarias, haciéndola una herramienta ligera, flexible y poderosa para este propósito [14].

JavaScript

Es un lenguaje de programación orientado a prototipos, lo que implica que los objetos pueden heredar directamente de otros objetos, en lugar de utilizar clases como en la programación orientada a objetos tradicional [15]

Angular:

Es una plataforma de desarrollo, que incluye un marco de trabajo basado en componentes que permite crear aplicaciones webs escalables, agregando bibliotecas integradas que cubren una amplia variedad de características como enrutamiento, administración de formularios, comunicación cliente-servidor y otras características [16].

Angular material:

Es una biblioteca de componentes de diseño basado en el lenguaje de diseño Material design, proporcionando una serie de componentes predefinidas para aplicaciones web y que se integran perfectamente con aplicaciones desarrolladas en angular [17]**.**

TypeScript:

Es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft, incluyendo nuevas características al lenguaje de programación JavaScript como clases, interfaces, y tipado estático, lo que ayuda a reducir errores en tiempo de ejecución y detectarlos en tiempo de compilación [18].

TailwindCSS:

Es un marco de trabajo de CSS que establece clases para el diseño de interfaces de usuario. A diferencia de otros Frameworks de css, tailwind no está enfocado en estilos visuales predefinidos, sino en clases que pueden ser utilizadas a conveniencia para la fácil personalización de los componentes [19].

Chart.js

Es una librería de JavaScript que permite crear gráficas estadísticas de manera interactiva con base a datos, incluyendo personalización, adaptables, Animaciones y la facilidad de integración [20].

NgRx

Es una gestión del estado global para aplicaciones hechas en angular y está inspirada en el patrón de diseño Redux. Diseñada para escribir aplicaciones consistentes y de alto rendimiento sobre angular [21]

Scikit-learn:

Es una librería gratuita de aprendizaje automático en Python, popular por la implementación de diversos algoritmos de aprendizaje automático, como clasificación, regresión, clustering, entre otros. Es usada en diferentes tipos de proyectos como el análisis de sentimientos, reconocimiento de voz, reconocimiento de imágenes y otros campos, convirtiéndola en una librería indispensable para trabajos de aprendizaje automático [22].

Smtlib

El módulo smtplib define un objeto de sesión de cliente SMTP que se puede utilizar para enviar correo a cualquier máquina de Internet con un demonio de escucha SMTP o ESMTP [23].

Zimbra

Es una plataforma de mensajería y colaboración que permite enviar y recibir correo electrónico, agendar citas y reuniones, crear contactos, organizar tareas y compartir documentos y archivos [24].

Vmware

VMware Workstation es una línea de productos de Hipervisor de escritorio que permiten a los usuarios ejecutar máquinas virtuales, contenedores y clústeres de Kubernetes [25].

Centos 7

Es una distribución basada en las fuentes de Red Hat Enterprise, donde su mayor implementación es el empresas y organizaciones de gran tamaño, y que estén destinadas al uso de servidores [26].

MongoDB

Es una base de datos NoSQL, orientado a documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado, permitiendo almacenar documentos en formato JSON o BSON [27].

Pymongo

Es una biblioteca de Python que proporciona una interfaz para trabajar con la base de datos MongoDB, es la forma recomendada para interactuar con la base de datos ya que ofrece todas las herramientas necesarias para realizar operaciones [28].

xlsx

Es una biblioteca alojada en npm para el lenguaje de programación TypeScript, enfocada en el trabajo de archivos de hojas de cálculo, al estar enfocada en TypeScript puede ser implementada junto con Angular [29].

jsPDF

Es una biblioteca que se encuentra en los repositorios de npm y se utiliza para generar documentos PDF en el lado del cliente [30].

1.3. Objetivos

1.3.1. Objetivos generales

Desarrollar un agente mediante machine learning basado en una técnica de clasificación de texto para la detección y prevención de correos electrónicos con contenido spam en el servicio de correo electrónico Zimbra para un GAD municipal.

1.3.2. Objetivos específicos

- Identificar un algoritmo de machine learning de clasificación de textos para la detección y clasificación de correos electrónicos como spam.
- Emplear agente en el servidor de correo electrónico Zimbra para la detección de spam en tiempo real.
- Diseñar medio alertivo para la rápida detección de ataques cibernéticos a través del servicio de correo electrónico.

1.4. Justificación del proyecto

El correo electrónico es una de las formas de comunicación más utilizadas en todo el mundo, y en los últimos años su uso ha ido en aumento. En 2021, se estima que había alrededor de 319.6 billones de usuarios de correo electrónico en todo el mundo, y se espera que este número haya aumentado a 333.2 billones en 2022. Además, se espera que el uso del correo electrónico siga creciendo en los próximos años, con una tasa de crecimiento del 4.1% hasta el 2025 [31].

Kaspersky llevó a cabo un estudio en 2021 sobre los usuarios que han recibido mensajes de phishing, y publicó que se evitaron alrededor de 25 millones de intentos de clics. Ecuador es el segundo país latinoamericano en recibir ataques de phishing con el 10.73% de usuarios afectados, lo que lo convirtió en el décimo país a nivel mundial [32].

País	Proporción de usuarios atacados (%) *
Brasil	12,39%
Francia	12,21%
Portugal	11,40%
Mongolia	10,98%
Reunión	10,97%
Brunei	10,89%
Madagascar	10,87%
Andorra	10,79%
Australia	10,74%
Ecuador	10,73%

Fig. 5. Porcentaje de usuarios Kaspersky cuyos equipos reaccionó al sistema de antiphishing [32].

El desarrollo del agente permitirá la monitorización en tiempo real de los correos electrónicos e identificar aquellos que contengan características de spam. Detectado un correo como spam, el agente reubicará el correo electrónico desde la bandeja de entrada hacia la carpeta de spam, enviando una alerta al usuario sobre la detección de este, además brindarle la opción de que el usuario pueda reportar el correo electrónico detectado de tal forma que pueda ser analizado por el administrador con el consentimiento previo del usuario y aplicar normas en el servicio de correo.

De esta manera al detectar y remover el correo electrónico detectado como spam, se ayudará a reducir la cantidad de correos electrónicos no deseados que llegan a la bandeja de entrada de cada usuario, evitando posibles fraudes o ataques informáticos, protegiendo la información sensible de la empresa y los trabajadores, añadiendo una capa extra de seguridad en el uso de este servicio.

Los resultados de la investigación servirán como punto de referencia para desarrollar nuevas herramientas de ciberseguridad que apliquen técnicas de aprendizaje automático y adaptarlos a otros tipos de servicios, esto contribuirá a mejorar la seguridad de la información de las empresas, protegiéndolas de posibles amenazas cibernéticas y garantizando la integridad y privacidad de los datos.

El tema propuesto está relacionado con los objetivos del plan de desarrollo, aplicando el eje social, detallando:

Objetivo 5. Proteger a las familias, garantizar sus derechos, servicios, erradicar la pobreza y promover la inclusión social [33].

Política 5.5. Mejorar la conectividad digital y el acceso a nuevas tecnologías de la población [33].

Lineamientos territoriales

Pol. 5.4.

A4. Fortalecer la conectividad y el acceso a las TIC como una vía para mejorar el acceso a otros servicios [33].

1.5.Alcance del proyecto

Este proyecto se enfocará en Zimbra, una plataforma de colaboración y correo electrónico de código abierto. Al utilizar esta solución, los usuarios tienen la libertad de personalizar y adaptar la aplicación según sus necesidades. Además, el enfoque en el software libre permite la colaboración con otros usuarios en la comunidad para mejorar la plataforma, añadir nuevas funcionalidades y solucionar errores.

El proyecto se centra en el desarrollo de un agente para la detección de correos spam en el servicio de correo electrónico de Zimbra, aplicando una técnica de aprendizaje automático que permita la clasificación de texto, con el fin proporcionar una capa de seguridad dentro de este servicio. El agente podrá detectar si un correo electrónico es spam, de serlo removerá el mismo de la bandeja de entrada hacia la carpeta de spam, además de enviar una alerta al usuario de la detección del correo y brindarle la opción de reportar el correo electrónico.

El agente no analizará la red, direcciones IPs, dominios de remitentes y receptores, cabeceras, asunto, archivos adjuntos, solo detectará si un correo es spam por medio del análisis del mensaje del correo electrónico, aunque el administrador tendrá la opción de agregar a usuarios a listas negras o listas blancas por medio de su correo electrónico para realizar un mejor filtrado.

Cabe recalcar que no se procederá a almacenar ningún tipo de información personal del usuario sin su consentimiento, tales como dominios de correos, nombres de usuarios, mensajes, y todo relacionado con el correo. Además, que el agente no eliminará el correo electrónico detectado, solo lo removerá a la carpeta de spam, pero seguirá siendo visible por el usuario.

El usuario antes de reportar el correo electrónico como spam, deberá leer los datos que se recopilarán del mismo para ser analizados por el administrador, si está de acuerdo con los términos entonces podrá reportarlo, caso contrario no se almacenará información confidencial de la comunicación entre usuarios.

Se añadirá una interfaz web en dónde se pueda observar datos y gráficas relacionados con la detección de spam, tales como la cantidad de correos electrónicos detectados por el agente, tanto normales como spam, las alertas enviadas a los usuarios y los correos que han sido reportados, de tal forma que pueda ver los detalles de este.

El agente solo detectará correos como spam cuando este se encuentre en ejecución, caso contrario no podrá realizarlo, además de no hacer un análisis del tráfico de la red, sino que se realizará un monitoreo constante de la carpeta en busca de posibles correos spam. Además, que el usuario solo podrá alertar un correo detectado como spam si el backend se encuentra en ejecución. Recalcando que el agente estará enfocado en una solución alertiva y no correctiva.

A continuación, se explicará el alcance de cada metodología seleccionada:

Metodología KDD

Selección

En esta etapa se seleccionarán los conjuntos de datos adecuados que servirán como base para el análisis y detección de correos electrónicos clasificados como spam. Este proceso implicará la selección de varios conjuntos de datos que contengan registros de correos electrónicos detectados como spam y correos normales. Se considerarán diversas fuentes y repositorios de datos que contengan conjuntos de datos previamente identificados como spam y normales.

Preprocesamiento/limpieza

En esta etapa se seguirán una serie de pasos para limpiar y organizar los conjuntos de datos obtenidos previamente, realizando una traducción del conjunto de datos al idioma español e identificar los atributos que no sean relevantes para la detección

de spam. Se aplicará la técnica de Tokenización para dividir los textos del conjunto de datos en unidades más pequeñas.

Transformación/reducción

Durante esta etapa se combinarán los diversos conjuntos de datos y formar uno solo conjunto de datos en formato csv. Este paso permitirá obtener un conjunto de datos consolidado sobre correos electrónicos normales y spam.

Minería de datos

Se entrenará a cada algoritmo de aprendizaje automático con el conjunto de datos previamente preprocesado y obtener los resultados de la implementación de cada algoritmo y conocer sus estadísticas por medio de tablas. La selección de algoritmos será por medio de fuentes bibliográficas de los algoritmos con mejor precisión para la detección de spam.

Interpretación/evaluación

Se llevará a cabo una evaluación de los resultados obtenidos en la fase anterior y se tomará la decisión de qué algoritmo se adapta mejor a la solución del problema. Esta evaluación incluirá el análisis de las métricas de evaluación del rendimiento de cada algoritmo para comprender su desempeño, luego se compararán los resultados de los diferentes algoritmos en términos de su eficacia y desempeño en la detección de spam. Finalmente, se seleccionará el algoritmo que presente los mejores resultados en base a estas métricas y se procederá a implementarlo en el agente encargado de la detección de correos electrónicos como spam. Esta etapa de evaluación garantizará la elección del algoritmo más adecuado y eficiente para resolver la resolución del problema planteado.

Metodología OMSTD

El agente constará de tres componentes, el script que será el encargado de detectar, remover el correo electrónico detectado como spam y alertar al usuario, seguido de una interfaz web que permita al administrador ver la cantidad de correos electrónicos detectados como normales y spam, además de observar los correos que

han sido reportados por los usuarios, por último, un backend que reciba estas peticiones del cliente y proporcione los datos desde el servidor.

Esta metodología será aplicada para cada componente siguiendo las siguientes descripciones:

Organización y estructura

En esta fase se realizará una comparativa del lenguaje de programación y marco de trabajo para determinar cuál es el más adecuado para el desarrollo de la herramienta, el objetivo de esta etapa es determinar la arquitectura de trabajo que se utilizaré en cada componente.

Entrada y salida de información

Se llevará a cabo la definición de los tipos de entrada y salida de información requeridos por el agente, frontend y backend.

Redistribución

Se realizará la programación de cada una de las partes que conforman la arquitectura del agente. Esto implica escribir el código necesario para cada componente, incluyendo el procesamiento de datos y la implementación de algoritmos de detección de spam.

El siguiente paso es implementar e integrar cada una de ellas. Esto implica poner en práctica el código desarrollado y asegurarse de que las diferentes partes del agente puedan comunicarse y compartir datos de manera efectiva.

Despliegue

Se llevarán a cabo actividades relacionadas con la configuración del entorno y la implementación de la herramienta de detección de spam.

Se realizará la configuración del entorno de despliegue, lo que implica establecer la infraestructura necesaria para la implementación de la herramienta incluyendo configuración de servidores de correo, servidores web locales y bases de datos.

Una vez que el entorno esté configurado, se procederá al despliegue de la herramienta en un entorno de prueba. Durante este proceso, se transferirán los

archivos y recursos necesarios al entorno seleccionado y se ejecutarán los pasos necesarios para poner en funcionamiento la herramienta.

Posteriormente será desplegado al servidor de correo electrónico Zimbra del GAD municipal, con el fin de poner a prueba el agente con datos reales.

Metodología CSF

Fase 1 Identificar

Entendimiento de la organización.

Durante esta fase, se llevará a cabo la recopilación de información sobre el servidor de correo electrónico, como las políticas de acceso, activos de información, la cantidad de usuarios con correo institucional, los dominios permitidos y bloqueados, así como el tipo de acceso al correo. Esto permitirá tener una comprensión completa del entorno de correo electrónico de la organización, permitiendo mejorar la identificación y evaluación de posibles riesgos de seguridad, para realizar este proceso se realizarán entrevistas y métodos de observación para recolectar la información necesaria.

Fase 2 Proteger

Se enfocará exclusivamente en el servidor de correo electrónico Zimbra y las medidas implementadas para salvaguardar sus activos. La investigación se centrará en identificar y analizar las políticas de gestión de identidad, autenticación y control de acceso que se aplican en el servicio de correo electrónico.

Fase 3 Detectar

El alcance de la recolección de información se centrará en los métodos y herramientas empleadas en el servidor de correo electrónico Zimbra para detectar correos electrónicos clasificados como spam. Se identificarán las tecnologías y algoritmos utilizados por Zimbra y de terceros para llevar a cabo la detección automática de mensajes no deseados.

Fase 4 Responder

Se recopilará información sobre los mecanismos implementados en el servidor de correo electrónico Zimbra para alertar a los usuarios o administradores sobre la recepción de correos electrónicos sospechosos o clasificados como spam.

Fase 5 Recuperar

Se identificarán los procesos establecidos para hacer frente a posibles fallos del sistema que pueden afectar el acceso o la disponibilidad de los correos electrónicos en Zimbra, además de los planes establecidos para enfrentar situaciones de ataques informáticos y como se recuperan los datos.

Niveles de implementación

Durante cada subcategoría de las cinco categorías de la metodología CSF, se procederá a realizar actividades para poder identificar los procesos de cada una de ellas y así determinar cuál es el nivel actual de cata subcategoría con relación al correo electrónico.

Perfiles

Identificados los niveles actuales de cada subcategoría, se procederá a crear un perfil objetivo con el fin de establecer los niveles de seguridad deseados.

1.6.Metodología del proyecto

1.6.1. Metodología de Investigación

Se utilizará la metodología de investigación exploratoria [34] para recopilar información de fuentes especializadas, estudios y revisión de trabajos relacionados con el tema propuesto. Esto permitirá obtener una comprensión detallada del problema y aplicar los métodos y herramientas adecuados para resolverlo de manera efectiva.

Para llevar a cabo la metodología de investigación diagnóstica, se realizarán entrevistas al administrador del servidor del GAD municipal para conocer la cantidad de correos que se transmiten diariamente y cuántos de ellos son considerados como spam, además de realizar métodos de observación para identificar características del servicio de correo electrónico.

1.6.2. Técnicas de recolección de información

Técnica

Estudio de observación y entrevistas.

Instrumentos

El estudio de observación será implementado en el departamento de TI del GAD municipal, con el objetivo de recolectar información respecto al servidor de correo para su posterior análisis

Las entrevistas serán dirigidas a las autoridades del departamento de TI, para conocer sobre el funcionamiento del servidor de correo electrónico y las normas de seguridad empleadas en el mismo.

Las fuentes especializadas servirán como análisis para el entendimiento de la problemática que se está investigando.

Población

- Autoridades del departamento de TI.

1.6.3. Beneficiarios del proyecto

Los beneficiarios serán el administrador del servicio de correo electrónico del GAD municipal, donde el algoritmo permitirá saber la cantidad de correos electrónicos detectados como spam en el servicio de Zimbra, y posteriormente al reporte realizado por el usuario, se procederá a analizar a detalle el correo electrónico detectado como spam y aplicar estrategias para mitigar este suceso.

Los beneficiarios indirectos serán los usuarios que poseen una cuenta de correo electrónico en el servicio de Zimbra, ya que el algoritmo les ayudará a evitar tener correos electrónicos clasificados como spam en la bandeja de entrada, pudiendo evitar caer en estafas por correos no deseados.

1.6.4. Variable

Se comparará la cantidad de correos electrónicos detectados como spam antes y después de su implementación del agente en el servicio de Zimbra.

1.6.5. Análisis de recolección de datos

Durante este proceso se aplicaron dos técnicas para poder recolectar información relevante sobre el funcionamiento del servicio del correo electrónico del GAD

municipal, tales como el método de observación y entrevistas realizadas al administrador del servicio de TI.

Duranta la técnica de observación aplicada para conocer el funcionamiento del panel de administración de Zimbra por parte del administrador del servicio detallado en el Anexo 2, nos da a conocer que el administrador hace uso de esta herramienta para monitorear el flujo de trabajo del servicio, donde se puede observar información como las sesiones activas y la cola de correos, además de notificar si un usuario ha ingresado de manera incorrecta el dominio de un destinatario, por lo que el servicio de Zimbra rechaza el envío.

Al finalizar el día, el administrador recibe un correo electrónico sobre el historial de uso del servicio, así mismo la cantidad de espacio en almacenamiento ocupado durante ese día y las cuentas de usuarios que más correos enviaron y recibieron, de esta forma se obtuvo que diariamente se envían aproximadamente 200 correos electrónicos a dominios internos y externos, determinando que este servicio es altamente usado por los trabajadores.

La técnica de observación aplicada para conocer el funcionamiento del servicio de correo electrónico Zimbra detallado en el Anexo 3, nos da a conocer que el servidor de correo electrónico se encuentra virtualizado con el entorno de Proxmox, donde se brinda servicio a más de 400 usuarios registrados con el dominio del servicio de correo electrónico del GAD. Actualmente se encuentran con problemas de spam dentro del servicio, donde en el Anexo 4 se puede identificar un dominio externo al GAD, asimismo, en el Anexo 5, en donde ambos tienen algo en común, y es que contienen dominios parecidos a entidades públicas, y en cada correo contienen enlaces que redireccionan hacia una página externa.

Para poder detectar si un correo electrónico es spam lo hacen mediante el reporte diario que entrega Zimbra, ya que se puede identificar todos los usuarios que recibieron y enviaron mensajes, pero este servicio tiene un sistema de detección de Spam limitado, ya que ambos ejemplos de correos electrónicos mencionados anteriormente llegaron a la bandeja de entrada del usuario y no fueron removidos a la carpeta de spam.

Existen segmentaciones de red en los diferentes departamentos del GAD, pero no existe una restricción en cuanto al acceso a páginas webs, lo que puede representar un riesgo de seguridad en la organización, ya que los usuarios pueden ingresar sus dominios de correo electrónico en páginas no confiables, lo que lleva a que este reciba correos con contenido spam.

Al ser una entidad pública, los correos electrónicos de los usuarios se encuentran públicos en la página oficial del GAD municipal, lo que conlleva a que los mismos puedan recibir correos electrónicos con contenido spam sin necesidad de que un usuario haya ingresado su nombre de correo en alguna página no confiable.

El servicio de Zimbra no cuenta con un filtrado de correos electrónicos categorizados como spam que reenvíe el mensaje detectado a su carpeta correspondiente, por lo que el mensaje llega directamente a la bandeja de entrada y llegar a ser potencialmente peligrosos. De igual forma, el administrador no cuenta con una forma automatizada de detectar si el servicio de Zimbra cuenta con correos categorizados como spam en tiempo real, y debe de esperar a que finalice el día para poder leer los resultados.

1.7.Metodología de desarrollo

1.7.1. Metodología KDD

La metodología KDD permite descubrir conocimiento útil a partir de un conjunto de datos [8], dónde se proponen las siguientes fases para obtener los mejores resultados posibles:

Selección.

En esta fase se seleccionan diferentes conjuntos de datos relacionados al problema a solucionar, en este caso conjunto de datos relacionados con correos electrónicos normales y spam.

Preprocesamiento/limpieza.

Se emplearán procesos para poder preprocesar el conjunto de datos seleccionado en la fase anterior, con el objetivo de seleccionar las características esenciales para la

evaluación, en este caso las partes importantes de un correo electrónico para poder predecir si un mensaje es normal o spam.

Transformación/reducción.

En esta fase, se procederá a formar un solo conjunto de datos comprendidos entre spam y normales. De esta manera se busca realizar el entrenamiento de los algoritmos de una forma simplificada.

Minería de datos.

En esta etapa se seleccionará algoritmos de aprendizaje automático para poder evaluar los mismos mediante el conjunto de datos obtenido en las fases anteriores.

Interpretación/evaluación.

En esta fase se obtendrá los resultados de los entrenamientos de cada algoritmo de aprendizaje automático, con el fin de identificar y seleccionar el algoritmo con mejor puntuación.

1.7.2. Metodología OMSTD

Es una metodología que proporciona una guía flexible y un marco de trabajo para el desarrollo de herramientas enfocadas en la seguridad informática [35], en la que se centra en varios aspectos, una de ella es la de conceptos de desarrollo que se dividen en varias etapas [36]:

Organización y estructura

Para desarrollar el agente de detección de spam, se siguió el modelo de programación modular, que se basa en dividir un problema complejo en subproblemas más pequeños [37], de tal manera que se puede modularizar el agente y obtener una mejor organización en cuanto a código se refiere.

Entrada y salida de información

El agente esperará como entrada de información una ruta de un archivo con extensión .msg, en la cual indica que es un archivo de correo electrónico que servirá para su posterior análisis y almacenar datos en la base de datos.

La salida de información del agente consistirá en la generación de alertas dirigidas al administrador del sistema. Estas alertas serán enviadas a través de correo electrónico y contendrán detalles relevantes sobre el análisis realizado por el agente

Redistribución

El agente podrá ser implementado solo en versiones de Zimbra que sean distribuciones Linux, ya que solo está disponible en este sistema operativo.

El agente se distribuirá en tres componentes, el agente que será ejecutado en el servidor de correo electrónico y detectará spam en los correos, el segundo componente será el ambiente web que permitirá gestionar los mensajes detectados como spam, y por último un backend que recibirá las peticiones del ambiente web para poder ser procesadas en el servicio Zimbra.

Despliegue

El agente será desplegado en un servidor de correo electrónico controlado, lo que permitirá simular un entorno real y la capacidad de analizar los correos electrónicos de manera efectiva. Este despliegue asegurará que el agente pueda aplicar su algoritmo de detección de spam y realizar el análisis correspondiente en tiempo real, brindando así una solución confiable y eficiente para la detección de correo no deseado.

Posteriormente se desplegará el agente en un servidor de correo electrónico del GAD.

1.7.3. Metodología CSF

Desarrollado por el instituto nacional de normas y tecnología (NIST), en la que cuenta con componentes principales como el Framework Core, Niveles de implementación y perfiles, donde las funciones de alto nivel del Framework Core son la identificar, proteger, detectar, responder y recuperar, de esta forma ayuda a las organizaciones de todo tamaño a comprender los riesgos de ciberseguridad [38].

Las fases que se harán uso son las siguientes:

Identificar: Se llevará a cabo actividades para identificar las normas de seguridad implementadas en el servidor de correo electrónico, con el objetivo de identificar los procesos que se sigue para poder identificar a un correo electrónico como spam

Proteger: Se realizarán actividades para conocer las medidas de protección que implementa el servidor de correo electrónico para proteger al servicio de correo electrónico, tales como la gestión de acceso al servicio.

Detectar: Las actividades a realizar serán para conocer si se emplean sistemas de monitoreo continuo para la detección de anomalías y eventos respecto al spam en el servicio de correo electrónico

Responder: En esta fase, se analizará las planificaciones de respuesta, análisis y mitigación que se emplea en el servidor de correo electrónico para hacer frente a eventos de spam.

Recuperar: Se analizará los procesos de recuperación de datos que se emplean el servidor de correo electrónico

En cada una de las categorías mencionadas anteriormente, se analizarán si se cumplen con lo establecido y se establecerá el perfil actual que se encuentra el servidor de correo electrónico con base a cada subcategoría (Anexo 14), de tal forma que establecido el perfil actual se podrá estimar el perfil objetivo de cada subcategoría (Anexo 15).

Capítulo II

2. Propuesta

2.1.Marco Contextual

2.1.1. GAD Municipal

El GAD municipal está comprometida con atender y satisfacer las necesidades de la ciudadanía en general, brindando una amplia gama de servicios a su comunidad. Con el objetivo de mejorar la calidad de vida de los habitantes, la empresa se enfoca en ofrecer soluciones eficientes y accesibles [3].

Cuenta con una estructura organizativa sólida que se compone de 47 departamentos y cuenta con un equipo de 711 empleados altamente capacitados y comprometidos con su labor [3].

2.1.2. Misión

Somos un gobierno local líder, que promueve el desarrollo humano sostenible, entregando a la comunidad servicios de calidad y calidez; con tal propósito desarrolla una gestión eficiente, transparente y participativa; contribuyendo de esta manera al bienestar material y espiritual de la colectividad [39].

2.1.3. Visión

El Gobierno Autónomo Descentralizado Municipal del cantón La Libertad, con la participación activa de la ciudadanía y la planificación articulada con los distintos o iguales niveles de gobierno, contribuirá a construir un modelo de desarrollo humano sostenible y equitativo, que privilegia la consecución del buen vivir; constituyéndose de esta manera, en el motor del progreso cantonal y provincial. Su Talento humano es solidario, altamente competitivo, honesto y comprometido con su institución y su cantón [39].

2.1.4. Marco Legal

2.1.4.1.Ley orgánica de protección de datos personales

Artículo 37.- Seguridad de datos personales

El responsable o encargado del tratamiento de datos personales según sea el caso, deberá sujetarse al principio de seguridad de datos personales, para lo cual deberá tomar en cuenta las categorías y volumen de la información personal, el estado de la técnica, mejores prácticas de seguridad integral y los costos de aplicación de acuerdo con la naturaleza, alcance, contexto y los fines del tratamiento, así como identificar la probabilidad de riesgos [40].

El responsable o encargado del tratamiento de los datos personales, deberá implementar un proceso de verificación, evaluación, valoración continua y permanente de la eficiencia, eficacia y efectividad de las medidas de carácter técnico, organizativo y de cualquier otra índole con la finalidad de mitigar de forma adecuada los riesgos identificados [40].

El responsable o encargado del tratamiento de datos personales deberá evidenciar que las medidas adoptadas e implementadas mitiguen de forma adecuada los riesgos identificados [40].

En otras medidas, se podrán incluir las siguientes:

1) Medidas de anonimización, seudonomización o cifrado de datos personales;
2) Medidas dirigidas a mantener la confidencialidad, integridad y disponibilidad permanentes de los sistemas y servicios del tratamiento de datos personales y el acceso a los datos personales, de forma rápida en caso de incidentes; y
3) Medidas dirigidas a mejorar la residencia técnica, física, administrativa, y jurídica.
4) Los responsables y encargados del tratamiento de datos personales podrán acogerse a estándares internacionales para una adecuada gestión de riesgos enfocada a la protección de derechos y libertades, así como para la implementación y manejo de sistemas de seguridad de la información o a códigos de conducta reconocidos y autorizados por la Autoridad de Protección de Datos Personales

Artículo 40.- Análisis de riesgo, amenazas y vulnerabilidades

Para el análisis de riesgos, amenazas y vulnerabilidades, el responsable y el encargado del tratamiento de los datos personales deberán utilizar una metodología que considere, entre otras:

1) Las particularidades del tratamiento;
2) Las particularidades de las partes involucradas; y,
3) Las categorías y el volumen de datos personales objeto de tratamiento.

Artículo 41.- Determinación de medidas de seguridad aplicables

Para determinar las medidas de seguridad, aceptadas por el estado de la técnica, a las que están obligadas el responsable y el encargado del tratamiento de los datos personales, se deberán tomar en consideración, entre otros [40]:

1) Los resultados del análisis de riesgos, amenazas y vulnerabilidades;

2) La naturaleza de los datos personales;
3) Las características de las partes involucradas; y,
4) Los antecedentes de destrucción de datos personales, la pérdida, alteración, divulgación o impedimento de acceso a los mismos por parte del titular, sean accidentales e intencionales, por acción u omisión, así como los antecedentes de transferencia, comunicación o de acceso no autorizado o exceso de autorización de tales datos.

2.1.4.2.Código orgánico integral penal, COIP

Sección sexta

Delitos contra el derecho a la intimidad personal y familiar

Artículo 178.- Violación a la intimidad

La persona que, sin contar con el consentimiento o la autorización legal, acceda, intercepte, examine, retenga, grabe, reproduzca, difunda o publique datos personales, mensajes de datos, voz, audio y vídeo, objetos postales, información contenida en soportes informáticos, comunicaciones privadas o reservadas de otra persona por cualquier medio, será sancionada con pena privativa de libertad de uno a tres años [41].

Artículo 232.- Ataque a la integridad de sistemas informáticos.

La persona que destruya, dañe, borre, deteriore, altere, suspenda, trabe, cause mal funcionamiento, comportamiento no deseado o suprima datos informáticos, mensajes de correo electrónico, de sistemas de tratamiento de información, telemático o de telecomunicaciones a todo o partes de sus componentes lógicos que lo rigen, será sancionada con pena privativa de libertad de tres a cinco años [41].

Con igual pena será sancionada la persona que [41]:

1. Diseñe, desarrolle, programe, adquiera, envíe, introduzca, ejecute, venda o distribuya de cualquier manera, dispositivos o programas informáticos maliciosos o programas destinados a causar los efectos señalados en el primer inciso de este artículo.

2. Destruya o altere sin la autorización de su titular, la infraestructura tecnológica necesaria para la transmisión, recepción o procesamiento de información en general.

Si la infracción se comete sobre bienes informáticos destinados a la prestación de un servicio público o vinculado con la seguridad ciudadana, la pena será de cinco a siete años de privación de libertad [41].

2.2.Marco Conceptual

2.2.1. Correo electrónico

Es un método de comunicación que permite enviar y recibir mensajes con múltiples destinatarios o receptores a través de redes informáticas con internet, donde, además de incluir texto, se pueden añadir archivos digitales como imágenes, documentos, videos, etc., permitiendo una comunicación instantánea y eficiente [42].

2.2.2. Software Libre

Se refiere al software que da libertad a los usuarios y la comunidad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Se basa en la idea de libertad y no se refiere al precio del software. Se puede entender el concepto de "libre" como el de "libertad de expresión" y no como algo gratuito [43]. Este tipo de software da la posibilidad de que el usuario tenga el control total del sobre el servicio, pudiendo adaptarlo a las necesidades propias o de la institución en donde se aplica [44].

2.2.3. Ham

Los correos electrónicos que son enviados de manera legítima y por remitentes confiables o conocidos son considerados como Ham, es decir, son correos electrónicos válidos para el receptor [45].

2.2.4. Correos no deseados o spam

2.2.4.1.Spam

El spam es un correo no solicitado que se envía automáticamente a múltiples direcciones al mismo tiempo, y es conocido popularmente como correo basura. A menudo se utiliza para fines de marketing, creando publicidad para productos, pero también puede ser utilizado por hackers para enviar programas malignos [46].

2.2.4.2.Tipos de spam

2.2.4.3.Correos electrónicos no deseados

El correo electrónico spam es el tipo más frecuente de spam en Internet. Se caracteriza por inundar la bandeja de entrada del usuario y distraerlo de los correos importantes. Por suerte, la mayoría de los clientes de correo electrónico ofrecen herramientas para informar, filtrar y bloquear la mayor parte de los mensajes de spam [47].

2.2.4.4.Spam como código malicioso

El spam de programa maligno es una forma de spam que incluye programas con código malicioso y que suele llegar a través de mensajes de correo electrónico no deseados o mensajes de texto. Este tipo de spam puede entregar varios tipos de malware, desde ransomware hasta troyanos y spyware [47].

2.2.5. Phishing

El phishing es un tipo de ataque cibernético que utiliza correos electrónicos disfrazados con técnicas de ingeniería social para engañar al destinatario haciéndole creer que el mensaje es importante, como una solicitud del banco o una nota de la empresa, y hacer clic en un enlace o descargar un archivo adjunto [48].

2.2.6. Protocolos del correo electrónico

2.2.6.1.IMAP

Protocolo de acceso a mensajes de Internet, es un estándar que los clientes de internet de correo electrónico hacen uso para obtener mensajes de correos electrónicos del servidor de correo electrónico mediante una conexión TCP/IP [49].

Este protocolo permite añadir banderas de mensajes para detectar qué mensaje se está viendo, capaz de recuperar el correo electrónico desde un servidor antes de descargarlo y facilita la descarga de archivos multimedia cuando estos son en gran volumen [49].

2.2.6.2.POP3

El protocolo de oficina de correos se utiliza para recuperar correos electrónicos desde un servidor de correo electrónico, siguiendo un modelo cliente-servidor donde el cliente debe obtener los mensajes desde el servidor, sin embargo no se

admite la organización de mensajes en el servidor mediante el uso de carpetas y no permite que el correo electrónico sea visto antes de descargarlo, mientras que no es necesario la conexión a internet, ya que el cliente de POP3 no verifica periódicamente si existen actualizaciones en el servidor [50].

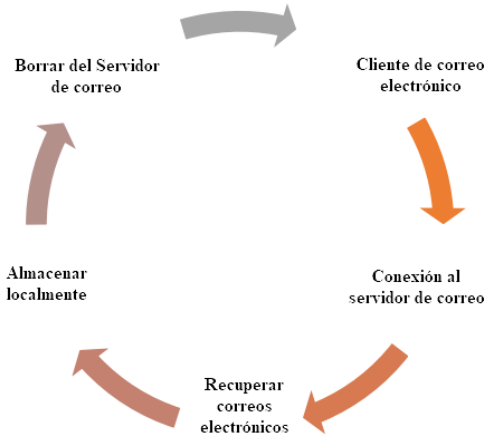


Fig. 6. Flujo de trabajo de recuperación de correo en POP3 [50].

2.2.6.3.SMTP

El protocolo simple de transferencia de correo es un proceso de cliente-servidor que sucede cuando el cliente en una red doméstica decide enviar un correo electrónico a una dirección fuera de la red, o cuando se envía un correo electrónico entre "hosts" en el mismo canal de comunicación, el modelo de SMTP se enfoca en transmitir y recoger mensajes a través de comandos entre el cliente y servidor [51].

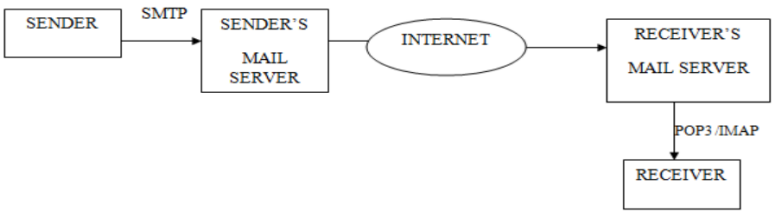


Fig. 7. Funcionamiento del protocolo SMTP [51].

2.2.7. Base de datos

Es una colección de datos organizados y estructurados, generalmente almacenados electrónicamente en un sistema informático, y gestionados por un sistema de administración de bases de datos (DBMS). La combinación de los datos, el DBMS y las aplicaciones asociadas se conoce como un sistema de base de datos [52].

2.2.7.1.Base de datos no relacionales

Las bases de datos NoSQL son una alternativa a las bases de datos relacionales, diseñadas para modelos de datos específicos y con esquemas flexibles que permiten un desarrollo rápido y escalabilidad horizontal. Se adaptan perfectamente a aplicaciones modernas que requieren bases de datos flexibles, escalables y de alto rendimiento. La alta funcionalidad y los tipos de datos específicos para cada modelo de datos hacen que las bases de datos NoSQL sean una solución atractiva para muchas aplicaciones [53].

2.2.7.2.MongoDB

Es una base de datos de documentos altamente escalable y flexible que utiliza un modelo de consultas e indexación avanzado. Con MongoDB, los desarrolladores pueden crear aplicaciones complejas y escalables con facilidad. La base de datos almacena datos en documentos flexibles similares a JSON, lo que permite cambios en la estructura de los datos con el tiempo. Además, MongoDB ofrece consultas ad hoc (dinámicas), indexación y agregación en tiempo real para acceder y analizar datos de manera potente. Es una base de datos distribuida en su núcleo, lo que permite una alta disponibilidad, escalabilidad horizontal y distribución geográfica fácil de usar [27].

2.2.8. Inteligencia artificial

La inteligencia artificial (IA) es un campo de la informática que busca simular las capacidades de inteligencia del cerebro humano. Se trata de un conjunto de tecnologías que permiten a las computadoras llevar a cabo tareas avanzadas, como procesar información visual, comprender y traducir lenguaje hablado y escrito, analizar grandes volúmenes de datos y ofrecer recomendaciones [54].

2.2.8.1.Aprendizaje profundo

El aprendizaje profundo (DL) forma parte de los métodos de aprendizaje automático e inteligencia artificial (IA) y se basa en redes neuronales artificiales (RNA) [55]. Destacando por su alto rendimiento en el manejo de grandes volúmenes de datos de seguridad, intentando emular el comportamiento del cerebro humano, permitiendo aprender a partir de grandes cantidades de información [55].

2.2.8.2.Aprendizaje automático

También conocido como aprendizaje automatizado o aprendizaje automático, es una rama de la inteligencia artificial enfocada en la implementación de técnicas que permiten a la computadora a aprender, lográndose mediate la aplicación de algoritmos que convierten datos a modelos o sistemas que puedan realizar tareas específicas [56]. El objetivo es desarrollar modelos y sistemas capaces de mejorar el desempeño y tomar decisiones acertadas basada en la experiencia adquirida a través de los datos [57].

2.2.8.2.1. Aprendizaje supervisado

Se basa en un modelo predictivo que utiliza un conjunto de datos previamente etiquetado y clasificado, para posteriormente entrenar al algoritmo seleccionado. El conjunto de datos se divide en un grupo de entrenamiento y de prueba, donde el algoritmo comparará los resultados obtenidos con los valores iniciales y poder clasificar las muestras, mejorando la estimación del resultado [58].

2.2.8.2.2. Aprendizaje no supervisado

Se caracteriza por la ausencia de un supervisor, utilizandose cuando se busca agrupar elementos de un conjunto según su similitud o medida de distancia. Este enfoque proporciona un análisis descriptivo implícito, ya que todas las piezas de información descubiertas por el algoritmo de agrupamiento contribuyen a una visión completa del conjunto de datos. El aprendizaje no supervisado es una forma en que el Machine Learning "aprende" datos no etiquetados, donde el algoritmo debe intentar comprenderlos por sí mismo, a diferencia del aprendizaje supervisado que se basa en conjuntos de datos etiquetados con una clave de respuestas [59].

2.2.9. Clasificación automática de texto

Es una rama de la inteligencia artificial ligada al desarrollo de máquinas de aprendizaje que se centra en el desarrollo de algoritmos capaces de analizar y categorizar grandes volúmenes de texto y que previamente hayan sido clasificadas por humanos [60].

2.2.10. Análisis de datos

El análisis de datos es un proceso que implica transformar datos recopilados en información significativa y útil. Mediante el uso de diversas técnicas, como el

modelado, se busca identificar tendencias y relaciones en los datos, lo que permite obtener conclusiones y perspectivas valiosas para respaldar el proceso de toma de decisiones [61].

2.2.10.1.Pandas

Es una librería código abierto para Python y se utiliza para trabajar con estructuras de datos de manera más eficiente, considerado una de las mejores librerías para el análisis de datos, ofreciendo una amplia variedad de funcionalidades comunes para la manipulación de datos, como seleccionar, filtrar por filas y columnas, cálculos estadísticos y combinación de datos [62].

2.2.10.2.Limpieza de datos

La limpieza de datos es el proceso de preparación de datos para su posterior análisis, en la que implica pasar por diversos filtros como eliminar, modificar, alterar datos incompletos, triviales, duplicados o que sigan un formato incorrecto, este proceso es útil para que durante el análisis de los datos no existan obstáculos en el proceso y generen resultados imprecisos [63].

2.2.11. Métricas de evaluación

2.2.11.1.Precisión

Es una métrica que mide la validez de los resultados estimados al analizar específicamente los casos clasificados como positivos y determinar si esas estimaciones fueron correctas, es decir, qué porcentaje de las predicciones positivas realizadas por el modelo son realmente positivas. Una alta precisión significa que el modelo tiene una alta capacidad para evitar clasificar incorrectamente ejemplos negativos como positivos [64].

2.2.11.2.Verdaderos positivos (Recall)

También conocido como sensibilidad, es una métrica fundamental en el campo del aprendizaje automático. Se refiere a la proporción de casos positivos reales que son correctamente identificados como positivos por el modelo. El recall mide la capacidad del modelo para capturar la mayoría de los casos positivos existentes, es decir, cuántos de los casos relevantes son realmente recuperados por el modelo [65].

2.2.11.3.F1-Score

Es una medida que tiene en cuenta tanto la precisión como el recall para calcular la puntuación general. Se puede interpretar como un promedio ponderado de los valores de precisión y recall, donde el F1-Score alcanza su valor máximo en 1 y su valor mínimo en 0. El F1-Score proporciona una evaluación equilibrada de la capacidad de un modelo de clasificación para mantener un equilibrio entre la precisión y el recall [66].

2.2.12. Algoritmos de aprendizaje automático para la detección de spam

2.2.12.1.Naïve Bayes

Es un algoritmo que se basa en la aplicación de bayes, donde su objetivo es asumir la presencia o ausencia de una característica en particular en una clase no esté relacionada con la presencia y ausencia de cualquier otra característica, siendo una técnica usada para predecir la probabilidad futura de que una característica pertenezca a una clase [67].

2.2.12.2.Support Vector Machine

Es un algoritmo de aprendizaje supervisado ampliamente utilizado en diversas aplicaciones, como procesamiento de señales médicas, procesamiento de lenguaje natural y reconocimiento de imágenes y voz. El objetivo principal de SVM es encontrar un hiperplano óptimo que pueda separar eficientemente dos clases diferentes de puntos de datos [68].

2.2.12.3.Random forest

Es un conjunto de árboles de clasificación o regresión no podados que se crean mediante la selección aleatoria de muestras de los datos de entrenamiento. En el proceso de construcción, se seleccionan características aleatorias para cada árbol. Las predicciones se obtienen mediante la agregación de las predicciones individuales de cada árbol, ya sea a través de un voto mayoritario en clasificación o el promedio en regresión. En general, el Random Forest muestra un rendimiento mejorado en comparación con los clasificadores de un solo árbol [69].

2.2.12.4.Redes Neuronales

Son modelos computacionales que se basan en el funcionamiento del cerebro humano, están compuestas por neuronas interconectadas, donde la información se propaga a través de los pesos o coeficientes de ajuste de las conexiones neuronales. Cada neurona en la red está conectada con otras neuronas, y la estructura y la fuerza de estas conexiones determinan el comportamiento global de la red. Los pesos o coeficientes de ajuste permiten que la red aprenda y se adapte a partir de los datos de entrenamiento [70].

2.2.13. Arquitectura cliente-servidor

Es un modelo de comunicación entre dos programas informáticos en el que el programa cliente realiza una o varias peticiones de servicio al programa servidor, en este modelo, el cliente inicia la comunicación y se utiliza para ejecutar programas y acceder a los datos almacenados en el servidor, mientras que el servidor es responsable de proporcionar el servicio solicitado por el cliente [71].

Fig. 8. Arquitectura cliente-servidor [71].

2.2.14. Api Rest

E una interfaz de programación que se adapta a los principios de la arquitectura REST. Esta arquitectura se basa en restricciones y recursos que pueden presentarse en formatos como JSON o XML, identificándose a través de URI (Uniform Resource Identifiers). La API REST utiliza el protocolo HTTP como medio de transporte para desarrollar servicios web que son ligeros, flexibles y escalables, brindando una forma eficiente de comunicación entre aplicaciones [72].

2.3.Marco Teórico

2.3.1. Uniendo la ciberseguridad y el aprendizaje automático

Según Kaspersky, “La ciberseguridad es la práctica de defender las computadoras, los servidores, los dispositivos móviles, los sistemas electrónicos, las redes y los datos de ataques malicioso” [73]. Mientras que IBM define a la ciberseguridad como “la práctica de proteger los sistemas más importantes y la información confidencial ante ataques digitales” [74].

El aprendizaje automático es una rama de la inteligencia artificial, donde se enfoca en el desarrollo de sistemas que sean capaces de mejorar su rendimiento o aprender a partir de los datos que se les proporcione [75].

Existen tres categorías del aprendizaje automático: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo [76]. El aprendizaje supervisado se enfoca en la predicción y tiene como objetivo la clasificación o predicción de resultados específicos de interés, como determinar si una foto contiene un gato u otro animal [77]. Por otro lado, el aprendizaje no supervisado utiliza datos no etiquetados para agrupar características similares, como extraer características relevantes de imágenes [78]. Finalmente, el aprendizaje por refuerzo no recibe instrucciones sobre las acciones a tomar en cuenta, sino que debe de descubrir a través de prueba y error cuáles acciones se puede brindar una mejor recompensa, siendo usado, en juegos de mesa o en controles de sistemas [79].

Al unir la ciberseguridad con el aprendizaje automático, los profesionales en esta área pueden desarrollar sistemas más eficientes y efectivos para proteger redes, sistemas y datos contra ataques cibernéticos, brindando una protección más sólida y adaptativa a una organización completa o un problema en específico, dónde se pueden evidenciar las siguientes ventajas [80]:

- **Seguridad de prueba completo de los modelos aprendizaje automático:** bajo el entrenamiento de un algoritmo de aprendizaje automático y el correcto uso de un conjunto de datos para permitir entrenar este mecanismo, se puede obtener predicciones y resultados correctos de lo que se planee analizar o detectar.

- **Menos intervención humana:** la mayoría de los procesos que se desee analizar se realizan a través de los modelos de aprendizaje automático implementados.
- **Escaneo y mitigación rápida:** mediante el entrenamiento de algoritmos específicos, se logra una eficiencia considerable para detectar la presencia de ataques informáticos, realizando escaneo y brindan respuestas rápidas en caso de cualquier señal de intrusos.

El aprendizaje automático es imprescindible en el campo de la ciberseguridad para la prevención de amenazas y fortalecimiento de las protecciones en organizaciones, mediante la detección de patrones, de tal forma que esta rama de la inteligencia artificial se puede emplear a la detección de diferentes tipos de amenazas [81], tales como:

- Detección de programa malignos.
- Detección y filtrado de spam.
- Detección de ataques de denegación de servicio.

2.3.2. Importancia de implementar un clasificador de correos electrónicos de spam con aprendizaje automático.

Un clasificador de spam es aquel algoritmo o sistema que se usa para identificar y categorizar de manera automática los mensajes que se transmiten a través de un correo electrónico, cuyo objetivo principal es separar esta información de la vista del usuario final [82].

Algunos de los tipos de filtros que pueden ser aplicados para detectar spam en un correo electrónico son: Filtros basados en resúmenes de contenido y filtros extraídos de la cabecera; este tipo de clasificaciones reduce el índice de falsos positivos, sin embargo, no cuentan con una gran efectividad [82]. Los filtros son aplicados para poder entrenar cualquier algoritmo de aprendizaje automático, ya que se encargan de procesar los datos y ayudan a eliminar información innecesaria o redundante, para poder preprocesar grandes cantidades de datos de correos electrónicos se pueden seguir las siguientes técnicas [83]:

- **Tokenización:** extraer las palabras en el cuerpo del mensaje.

- **Lematización:** reducir las palabras a su estado base.

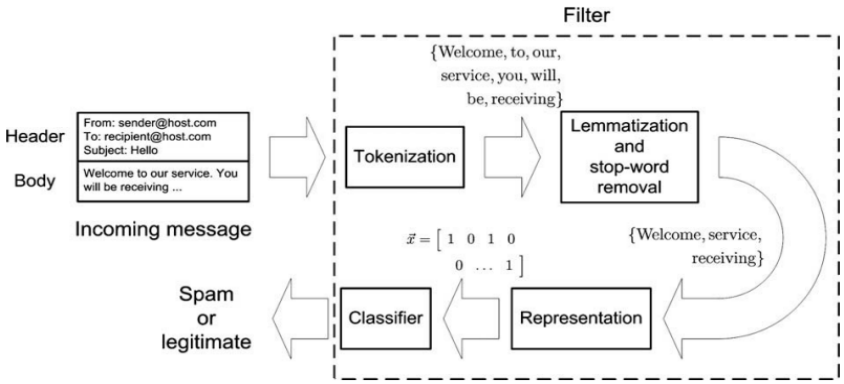


Fig. 9. Técnicas para preprocesar un correo electrónico previo al entrenamiento de un algoritmo [83].

2.3.3. SQL o NoSQL, ¿Por qué y qué base de datos usar en aprendizaje automático?

Las bases de datos son una herramienta esencial para la gestión de actividades digitales, ya que permiten la organización, mantenimiento y evaluación de información crucial para una organización, garantizando que la información se mantenga de manera estructurada y accesible [84]. Las ventajas mencionadas en esta publicación sobre el uso de las bases de datos en el aprendizaje automático son: simplicidad, minimizar el tiempo, fácil producción, eficiencia en la recuperación de los datos y la escalabilidad [85].

El lenguaje SQL desempeña un papel fundamental en la gestión de datos, especialmente cuando se trata de datos estructurados, particularmente en los sistemas de administración de bases de datos relacionales, donde el uso de este lenguaje propone variedad de acciones, que incluye la inserción, consulta, eliminación y actualización datos, además de la capacidad de controlar el acceso a la información [86].

En cambio, NoSQL se refiere a aquellas bases de datos que no siguen la modelo tradicional llamada bases de datos no relacionales, el término NoSQL se refiere a "no solo SQL" siendo una alternativa a este modelo y no un descarte de este [87].

Algunas razones importantes que se debe de considerar para seleccionar una base de datos no relacional sobre una relacional son las siguientes características [88]:

- Evitar complejidad innecesaria.
- Alto rendimiento.
- Escalabilidad horizontal
- Hardware de bajo costo

El análisis cualitativo realizado por Mayur, *et al.* [89] sobre el rendimiento de una base de datos NoSQL vs SQL, aborda la evaluación de rendimiento entre las bases de datos MongoDB y MySQL en operaciones de inserción y recuperación de datos, dónde empleó técnicas de balanceo de carga y particionamiento, destacando las ventajas que ofrece la base de datos no relacional, obteniendo los siguientes resultados.

Tabla 1 - Resultados por tiempo de inserción y recuperación en MySQL y MongoDB [89].

Prueba #	**Número de registros VS Tiempo empleado**		
	Número de registros	***MySQL (tiempo en segundos)***	***MongoDB tiempo en segundos)***
1	10	0.0511	0.005
2	20	0.0520	0.007
3	30	0.0566	0.01
4	40	0.0598	0.01
5	50	0.0698	0.01

2.3.4. Algoritmos de aprendizaje automático para la clasificación de correos electrónicos spam.

Los algoritmos de aprendizaje automático cumplen un rol importante en la clasificación de correos electrónicos como spam, ya que son capaces de procesar grandes volúmenes de datos y aprender patrones a partir de ellos, convirtiéndose en herramientas imprescindibles para la detección automatizada de correos

electrónicos como spam, permitiendo adaptarse y evolucionar a medida que los patrones y características de los correos electrónicos cambien [90].

Dentro de las categorías del aprendizaje automático: aprendizaje supervisado, y aprendizaje no supervisado [76], cuentan con una variedad de algoritmos para su categoría correspondiente, abarcando distintos tipos de problemas y escenarios [90]:

- **Aprendizaje supervisado**

  Red Neuronal Artificial: son sistemas que procesan información y están diseñados siguiendo la estructura y funcionamiento de las redes neuronales biológicas. Están diseñadas para el procesamiento en secuencia, aprendiendo de ejemplos y realizando predicciones futuras[91].

  Naïve Bayes: basado en el teorema bayesiano, implica realizar cálculos probabilísticos con el fin de encontrar la mejor clasificación para un conjunto de datos en un campo específico [92].

  Árbol de decisión: técnica estadística para predecir resultados y tomar decisiones basadas en observaciones [93].

- **Aprendizaje no supervisado**

  Agrupación K-means: usado en agrupaciones de datos y se basa en una medida de distancia para dividir el conjunto de datos en grupos llamados clústeres. Su objetivo principal es encontrar el centro de cada clúster, representado por el vector medio de los atributos numéricos o el vector modal de los atributos nominales de todas las instancias en ese clúster [94].

2.3.5. Precisión de algoritmos de aprendizaje automático aplicando conjunto de datos con correos electrónicos como spam y normal

El estudio realizado en el trabajo “Clasificación de spam mediante aprendizaje automático y procesamiento del lenguaje natural” [95] aplicaron diferentes algoritmos de aprendizaje automático, incluyendo el Naive Bayes y Máquina de vectores soporte, además de verificar la eficiencia de cada uno aplicando conjuntos de datos disponibles en Kaggle y Enron.

Tabla 2 - Comparación de la precisión de los algoritmos de aprendizaje automático [95].

Algoritmo		Naive Bayes (%)	Máquina de vectores soporte (%)	Árbol de decisión (%)	Bosque aleatorio (%)	K Vecino más próximo (%)			
	0/1					K=1	K=3	K=6	K=10
Precisión	0	95	95	91	96	95	93	90	89
	1	97	97	83	99	99	100	100	100
Recuperación	0	100	100	99	100	100	100	100	10
	1	68	86	41	74	66	50	30	24
F1 Score	0	97	99	95	98	97	96	95	94
	1	80	92	55	85	79	67	46	38
Exactitud		95.48	97.83	90.90	96.43	95.25	93.25	90.58	89.69

Las pruebas realizadas en este estudio fue el envío de cien correos electrónicos a una cuenta de Gmail y determinar si se clasificaban correctamente, obteniendo los siguientes resultados [95]*:*

Tabla 3 - Resultados del envío de cien correos electrónicos [95].

Característica	**Cantidad**
Número total de correos enviados	100
Emails Spam enviados:	57
Correos electrónicos normales enviados:	53
Correos electrónicos no deseados clasificados como spam (Verdadero positivo):	53

2.4.Requerimientos

Para el desarrollo e implementación del algoritmo dentro del servidor de correo electrónico, se necesitan los siguientes requerimientos:

R1. Para la implementación del agente, se necesitan las siguientes características en el servidor:

Tabla 4 - Requisitos de hardware para la implementación del agente

Característica	**Mínimo**	**Recomendado**
Sistema Operativo	Linux	Linux
Distribución	Indistinto	Indistinto
Memoria RAM	8 GB	16GB
Almacenamiento	1TB	5TB

R02. Para asegurar el adecuado funcionamiento del agente en el servidor, serán requeridas las siguientes características en términos de software.

Tabla 5 - Requisitos de software para el correcto funcionamiento del agente.

Característica	**Descripción**
Plataforma	Zimbra
Python	V3
PIP	V3
MongoDB	V4.4

R03. El administrador del servidor debe de crear una cuenta en Zimbra llamada "detected", que se encargará de enviar las alertas a través de este medio hacia el usuario administrador.

R04. El agente podrá ser ejecutado en el servidor si se cuenta con las siguientes librerías:

Tabla 6 - Librerías indispensables para la ejecución correcta del agente.

Librería	**Descripción**	**Versión**
watchdog	Detecta cambios en una ruta.	V.2.2.1

Librería	Descripción	Versión
smtplib	Crea sesión SMTP para enviar correos a cualquier dominio.	V.0.1.1
mailparser	Permite desestructurar y obtener las características de un correo electrónico (From, To, Body).	V.3.15.0
scikit-learn	Herramienta que incluye algoritmos de aprendizaje automático	V.0.24.2
PyMongo	Permite usar MongoDB a través de Python.	V.4.1.1
Cryptography	Permite trabajar con criptografía en Python	V.40.0.2
Pandas	Permite manipular datos	V.1.1.5
PyJWT	Permite crear tokens de JWT en Python	V.1.4.2
secure-smtplib	Permite enviar mensajes de correos electrónicos automatizados	V.0.1.1
Bcrypt	Permite el uso seguro de contraseñas	V. 4.0.1
Flask	Permite crear aplicaciones webs de manera sencilla con Python	V.2.0.3

R05. Al momento de implementar el agente en el servidor, se deberá volver a entrenar el modelo de aprendizaje automático seleccionado.

R06. Para poder ejecutar la interfaz web, se recomienda tener las siguientes características en un computador de escritorio.

Tabla 7 - Características recomendadas para la ejecución de la interfaz web.

Característica	**Mínimo**	**Recomendado**
Procesador	i7 de Octava Generación	i7 de Décima Generación
Memora RAM	4GB	8GB
Almacenamiento	256GB	1TB

R07. Para habilitar la ejecución de las API REST en el servidor para facilitar la gestión del panel de control, se requerirán las siguientes librerías externas.

Tabla 8 - Librerías para ejecutar el backend que gestionará los correos.

Software	**Versión**
Python	V3.10.0
Flask	V2.2.2
Flask-Cors	V3.0.10
PyJWT	V2.6.0
bcrypt	V4.0.1

R08. Para poder ejecutar la interfaz web, el computador a usar debe de tener los siguientes entornos de desarrollo.

Tabla 9 - Softwares para ejecutar la interfaz gráfica.

Software	**Descripción**
NodeJS	V16.17.0
Angular	V14.2.0

R09. Para poder visualizar la interfaz web, se recomienda usar los navegadores Google Chrome, Mozilla Firefox o Microsoft Edge.

R10. Dentro del servidor de correo electrónico, se requerirá que se habilite un puerto para poder ejecutar el Backend de la interfaz web.

R11. El servidor deberá de tener instaladas las siguientes características para ejecutar el backend:

Tabla 10 - Softwares para ejecutar el backend.

Software	Descripción
Python	V3
PIP	V3
Flask	V2.2.2.

R12: El administrador de TI deberá de ejecutar el script que contiene el agente para que este empiece a detectar correos e identificar posibles spams.

R13. Las notificaciones enviadas a través de la cuenta "detected" se deben enviar al instante en que se detecte un correo como spam.

R14. El administrador podrá ejecutar el ambiente web cuando crea conveniente de hacerlo para la gestión de los correos electrónicos.

2.5.Desarrollo de la propuesta

2.5.1. Metodología KDD

2.5.1.1.Selección.

Durante esta fase, se tomó en cuenta varios conjuntos de datos provenientes de plataformas que ofrecen estos datos en archivos ".csv", se tomaron en cuenta las siguientes plataformas para obtener el conjunto de datos:

Durante esta fase, se consideraron varias plataformas que brinden conjuntos de datos que contengan correos electrónicos como spam y que ofrezcan en formatos específicos, en este caso el "csv", entre las plataformas utilizadas están Kaggle y Github.

Los conjuntos de datos seleccionados que cumplan el formato de "csv" son:

- Fraud Email Dataset es un conjunto de datos en formato "csv" que contiene dos columnas de contenido, la primera es la del contenido, es decir el texto

de un correo electrónico, seguido de otra columna conocida como etiqueta que especifica si el texto es spam o no [96].

Text	Class
Supply Quality China's EXCLUSIVE dimensions at Unbeatable Price.Dear Sir, We are pleased to	1
over. SidLet me know. Thx.	0
Dear Friend,Greetings to you.I wish to accost you with a request that would be of immense benef	1
MR. CHEUNG PUIHANG SENG BANK LTD.DES VOEUX RD. BRANCH,CENTRAL HONG KON	1
Not a surprising assessment from Embassy.	0
Monica -Huma Abedin <Huma@clintonemail.com>Tuesday June 29 2010 6:01 AM'hanleymr@st	0
Pis print.H <hrod17@clintonemail.com>Thursday October 8 2009 8:01 PM'JilotyLC@state.gov'F	0
Dear Tom--H <hrod17@clintonemail.com>Friday December 11 2009 5:41 PMCould we schedule	0

Fig. 10. Imagen referencial del conjunto de datos "Fraud Email Dataset" [96].

Enron-spam es un conjunto de datos proveniente de la empresa Enron, la cual contiene un conjunto de datos subdivididas en carpetas llamadas ham y spam, pero contiene los mismos en formatos ".txt" [97].

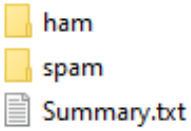


Fig. 11. Imagen referencial de conjunto de datos enron con su subconjunto de carpetas [97].

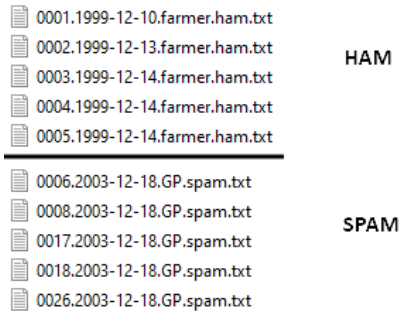


Fig. 12. Imagen referencial de los formatos de los subdirectorios del conjunto de datos Enron [97].

- Enron en formato ".csv" es un conjunto de datos proveniente del conjunto de datos de Enron, pero en esta ocasión con el formato adecuado y se encuentra disponible en el repositorio de GitHub [98].

Message ID	Subject	Message	Spam/Ham	Date
0	christmas tree farm pictures		ham	1999-12-10
1	vastar resources , inc .	gary , production from the high island...	ham	1999-12-13
2	calpine daily gas nomination	- calpine daily gas nomination 1 . doc	ham	1999-12-14
3	re : issue	fyi - see note below - already done	ham	1999-12-14
4	meter 7268 nov allocation	fyi	ham	1999-12-14
5	mcmullen gas for 11 / 99	jackie ,...	ham	1999-12-14

Fig 13. Imagen referencial del conjunto de datos enron en formato csv [98].

Los conjuntos de datos seleccionados bajo el formato de archivos csv son los siguientes:

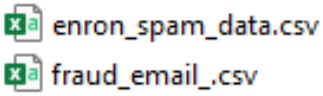


Fig. 14. Conjuntos de datos a utilizar en el desarrollo del proyecto.

2.5.1.2.Preprocesamiento/limpieza.

Eliminar datos duplicados

En esta etapa se procedió a eliminar los datos duplicados en cada conjunto de datos, ya que puede afectar negativamente al rendimiento y la eficiencia de los análisis posteriores. Al eliminar los datos duplicados, se obtiene la optimización del conjunto de datos al reducir su tamaño y eliminación de redundancia.

El siguiente código se ejecutó por cada conjunto de datos seleccionado en la fase anterior para poder realizar este proceso, obteniendo como parámetro la ruta del archivo a realizar el proceso, eliminando al mismo los archivos duplicados y creando un nuevo archivo, pero con los datos ya procesados, listo para realizar el siguiente proceso.

```python
import pandas as pd

def remove_duplicates_csv(ruta, archivo):
    dataset = ruta + archivo
    df = pd.read_csv(dataset)

    # Eliminar duplicados basados en la columna 'Text'
    df = df.drop_duplicates(subset='Message', keep='first')

    # Guardar el DataFrame modificado en un nuevo archivo CSV
    # en la subcarpeta 'no_duplicate/'
    df.to_csv(ruta + 'no_duplicate/' + archivo, index=False)

ruta = '[redacted]'
archivo = 'enron_spam_data.csv'
remove_duplicates_csv(ruta, archivo)
```

Fig. 15. Código para eliminar contenido duplicado en un archivo csv.

El fragmento de código mostrado en la siguiente imagen muestra la cantidad de los datos a través de sus columnas, de esta manera se obtiene información sobre el total de datos.

```python
import pandas as pd

def print_count_all_values(ruta, archivo):
    dataset = ruta + archivo # Ruta del archivo CSV
    df = pd.read_csv(dataset) # Leer el archivo CSV
    print(df.count())

ruta = '[redacted]'
archivo = 'enron_spam_data.csv'
print('Valores no preprocesados: ')
print_count_all_values(ruta, archivo)
print('-------------------------')
print('Valores preprocesados: ')
print_count_all_values(ruta + 'no_duplicate/', archivo)
```

Fig. 16. Función para mostrar el conteo de los valores de cada columna del archivo csv.

La salida de información de la ejecución del código descrito en la Figura 15, muestra la cantidad de los datos totales que se encuentran en cada columna del conjunto de datos en el archivo

```
Valores no preprocesados:
Text     11928
Class    11929
-------------------------
Valores preprocesados:
Text     10249
Class    10250
```

Fig. 17. Resultado del conteo de datos totales del conjunto de datos fraud_email no y preprocesado.

Como resultado se obtuvo una reducción de los datos, y a la vez, disparidad en las columnas “Text” y “Class” entre el conjunto de datos no procesado y procesado, teniendo como resultados en ambos conjuntos de datos que la columna “Class”

cuenta con una fila más que la columna “Text”, dando indicios que existen datos nulos.

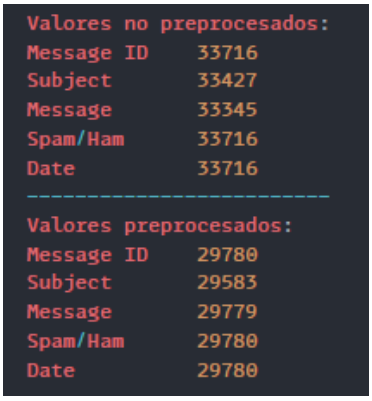


Fig. 18. Resultado del conteo de datos totales del conjunto de datos enron_spam_data.csv no y preprocesado.

Como resultado se obtuvo reducción de los datos, y también, una desigualdad en la cantidad de filas que se tiene en el conjunto de datos, donde en las columnas de “Subject” y “Message” tiene una diferencia de 4 dilas, donde se refleja una existencia de datos nulos entre estas características.

Eliminación de características del conjunto de datos

En la figura 8, el conjunto de datos cuenta con las columnas de “Text” y “Class”, siendo datos suficientes para el entrenamiento del algoritmo y no necesita pasar por este proceso de eliminación de características.

En la figura 11, se puede apreciar todas las columnas que cuenta el conjunto de datos “enron”, teniendo como características:

- Message ID
- Subject
- Message
- Spam/ham
- Date

Algunas de estas características no son relevantes para el entrenamiento del algoritmo, por ejemplo, los campos “Message ID”, “Subject” y “Date” no tienen

información relevante donde se pueda determinar si un correo electrónico es spam o no, por esta razón se procederá a eliminar estas características del conjunto de datos.

La siguiente función elimina las características mencionadas anteriormente y las almacena en un nuevo archivo con extensión .csv.

```python
import pandas as pd

def remove_column(ruta,archivo):
    dataset = ruta + archivo
    df = pd.read_csv(dataset)
    df = df.drop(['Message ID'], axis=1)
    df = df.drop(['Subject'], axis=1)
    df = df.drop(['Date'], axis=1)
    df.to_csv(ruta  + archivo, index=False)


ruta = '[redacted]'
archivo = 'enron_spam_data.csv'
remove_column(ruta,archivo)
```

Fig. 19. Función que elimina columnas de un archivo csv.

La figura 18 refleja los resultados de la eliminación de las columnas que no serán procesadas para el entrenamiento del algoritmo.

Message	Spam/Ham
	ham
gary , produ...	ham
- calpine dail...	ham
fyi - see note...	ham
fyi	ham
jackie ,...	ham

Fig. 20. Nuevo conjunto de datos aplicando la función de eliminar columnas.

En el caso del conjunto de “datos fraud_email.csv” no se hará este procesamiento, ya que solo cuenta con las columnas de “Text” y “Class”, teniendo solo lo necesario.

Eliminar datos nulos

En la etapa de eliminación de datos duplicados, se pudo evidenciar que existía una desigualdad en la salida de información entre las columnas “Text” y “Class”, dando

como resultados de 11928 y 11929 respectivamente, donde se daba una señal de existencia de datos nulos en la columna “Text”.

La siguiente funcionalidad elimina los datos que cuenten con campos nulos, específicamente que cuenten con la característica “NaN”, de esta forma se puede obtener un conjunto de datos columnas equilibradas en términos de cantidad de datos.

```python
import pandas as pd

def remove_nulls_csv(ruta, archivo):
    dataset = ruta + archivo

    df = pd.read_csv(dataset)
    print('Valores preprocesados con valores nulos: ')
    print(df.count())

    df = df.dropna() # Eliminar filas con valores nulos
    df.to_csv(ruta  + archivo, index=False)
    print('-------------------------')
    print('Valores preprocesados sin valores nulos: ')
    print(df.count().to_string())
ruta = '[redacted]'
archivo = 'final_dataset.csv'
remove_nulls_in_csv(ruta, archivo)
```

Fig. 21. Función para borrar valores nulos dentro del conjunto de datos.

Resultados de la ejecución de la función que elimina los datos nulos del conjunto de datos “fraud_email.csv”, donde se muestra un antes y después de la ejecución.

```
Valores preprocesados con valores nulos:
Text     10249
Class    10250
-------------------------
Valores preprocesados sin valores nulos:
Text     10249
Class    10249
```

Fig. 22. Resultados del proceso para eliminación de datos nulos del conjunto de datos fraud_email.csv.

Resultados de la ejecución de la eliminación de los datos nulos en el conjunto de datos “enron_spam_data.csv”, detallando un antes y después de la cantidad de datos de cada columna, mostrando una igualdad.

```
Valores preprocesados con valores nulos:
Message     29779
Spam/Ham    29780
-------------------------
Valores preprocesados sin valores nulos:
Message     29779
Spam/Ham    29779
```

Fig. 23.Resultados del proceso para eliminación de datos nulos del conjunto de datos enron_spam_data.csv.

Traducción del conjunto de datos al idioma español

Una vez concluida la etapa de eliminar datos nulos, se pudo evidenciar las cantidades totales de cada conjunto de datos, teniendo como resultado lo siguiente:

fraud_email: 10249

enron_spam_data: 29779

En la funcionalidad descrita en la figura 22, traduce el campo “Text” del conjunto de datos “fraud_email” al idioma español, para lograr aquello se hace uso de la librería “Translator” que ayudará a realizar estas traducciones, además de la librería pandas que permitirá manipular el archivo csv, de tal forma que se lea los datos de la ruta original, y cuando es traducida se lo almacene en un nuevo conjunto de datos traducido al español.

```python
import pandas as pd
from googletrans import Translator

translator = Translator()

def traduction(file, languaje, destination):
    try:
        df = pd.read_csv(file)
        for i in range(0, len(df['Text'])):
            try:
                df['Text'][i] = translator.translate(df['Text'][i], dest=languaje).text
                df.to_csv(destination, index=False)
                print('Guardado: ', i)
            except:
                df.to_csv(destination, index=False)
                print('Error: ', i)
    except:
        print('Error al leer el archivo')

file = '[redacted]'
destination = '[redacted]/fraud_email_es.csv'
traduction(file,'es',destination)
```

Figura 1: Código para traducir el campo "Text" del conjunto de datos fraud_email.

Como salida de información del código ejecutado previamente, se obtiene un conjunto de datos con los campos de “Text” traducidas al español.

Text	Class
Sí, Iona lo tiene en la lista.	0
Información requerida. 1) Su nombre completo y dirección 2) Su número de teléfono privado,...	1
Burns esto va a pasar. Estamos comprometidos con su trabajo y otros documentos como ella....	0
ESTIMADO SEÑOR/Señora, MI NOMBRE ES INGENIERO. TONY EDEM EL DIRECTOR GENERAL DE...	1
Señora Secretaria: Gracias por comunicarse con la Secretaria Solís y convencerla de unirse a la...	0
Para tu información	0
Estimado señor o señora: Estamos muy interesados en establecer una nueva fábrica para pro...	1
8:00 am Llamada lv/primer ministro israelí Netanyahu8:10 am Residencia privada8:15 am SALID...	0
Abedin Huma <AbedinH@state.gov>Miércoles 15 de julio de 2009 13:44Re: ArtEstará fuera d...	0
Ver NIMills Cheryl D <MillsCD@state.gov> Sábado 11 de diciembre de 2010 1:36 PMFw: S está...	0
Aquí está el borrador	0
De acuerdo	0
ESTA ES UNA NOTIFICACIÓN OFICIAL DE FONDOS DEPOSITADOS A SU NOMBRE Y NO ES UNA...	1

Fig. 24. Conjunto de datos fraud_email traducido al español

Para poder traducir este conjunto de datos se hizo una comparativa entre el tiempo aproximado para poder completar esta acción, obteniendo los siguientes resultados.

Tabla 11 - Tiempo total estimado empleado en la traducción de conjunto de datos fraud_email.

Cantidad de campos a traducir	**Tiempo estimado de traducción en segundos**	**Tiempo estimado de traducción en horas y minutos**
1	2 seg	0.0005556 horas
10249	20498 seg	5 horas y 41 minutos

La comparativa detalló que se requirió aproximadamente 5 horas y 41 minutos, teniendo esto en cuenta se procedió a realizar esta misma tabla, pero tomando en cuenta el conjunto de datos de enron, lo cual en la tabla 12 se detalla los resultados estimados:

Tabla 12 - Tiempo estimado en la traducción de conjunto de datos enron.

Cantidad de campos a traducir	**Tiempo estimado de traducción en segundos**	**Tiempo estimado de traducción en horas y minutos**
1	3 seg	0.00083333 horas
29779	89337 seg	24 horas y 49 minutos

Se puede evidenciar que traducir el conjunto de datos Enron puede tardar hasta un día de espera porque la cantidad es relativamente grande respecto al conjunto fraud_email, por esta razón se procedió a dividir el primero en 5 partes iguales para poder traducir cada fracción.

La siguiente función lee un archivo csv, en este caso el conjunto de datos enron y lo divide en partes iguales, para eso se pasa como número de partes como valor de 5 que equivale cinco archivos distintos en partes iguales.

```python
import pandas as pd
import math

def separate_csv(file, num_parts, output):
    df = pd.read_csv(file)
    total_rows = len(df)
    rows_part = math.cell(total_rows / num_parts)

    # Dividir en DF en partes casi iguales
    parts = [df[i:i + rows_part] for i in range(0, total_rows, rows_part)]

    # Guardar cada parte en un archivo CSV diferente
    for i,part in enumerate(parts):
        part.to_csv(f'{output}_part{i}.csv', index=False)

file  = '[redacted]enron_spam_data.csv'
num_parts = 5
output = '[redacted]/partes_enron/enron_spam_data'
separate_csv(file, num_parts, output)
```

Fig. 25. Separación en partes del conjunto de datos enron.

El resultado de la ejecución previa es el conjunto de dato es el siguiente:

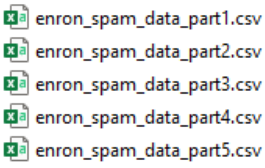


Fig. 26. Separación en partes iguales del conjunto de datos enron.

Una vez separado el conjunto de datos en 5 partes, se procedió a programar una función donde se pueda ejecutar cada parte en un hilo diferente del computador, así poder reducir el tiempo de espera para poder traducir el conjunto de datos completamente.

Para poder dividir el conjunto de datos en cinco partes diferentes se tuvo en consideración la cantidad de hilos de procesamiento del computador donde se estaba desarrollando la solución.

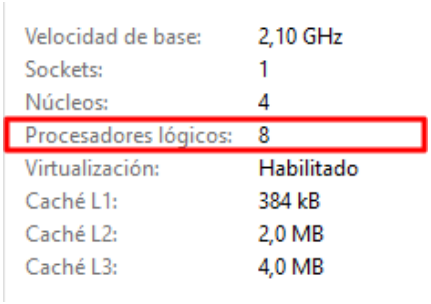


Fig. 27. Cantidad de hilos de procesamiento disponibles.

En la figura 26, se resalta que solo se contaba ocho hilos de procesamientos disponibles, donde se seleccionaron cinco hilos exclusivamente para la traducción del conjunto de datos, dejando como libre dos hilos ya que uno se encuentra ocupado en sí por el sistema operativo.

Traducción del conjunto de datos a español empleando en hilos de procesamiento

Para poder reducir el tiempo de traducción en el conjunto de datos Enron, se procedió a dividirlo en partes teniendo en cuenta la cantidad de hilos de procesamientos disponibles.

Se procedió a almacenar las rutas originales de las partes de los archivos en variables, seguido de su ruta de destino, para evitar escribir cada ruta completa se procedió a crear una ruta base para que las otras puedan "heredarla" y reducir código.

```
file_base = '                                                    /enron_spam_data_part'
destination_base = '                                                    /enron_spam_data_part'

file1 = file_base + '1.csv'
destination1 = destination_base + '1_es.csv'

file2 = file_base + '2.csv'
destination2 = destination_base + '2_es.csv'

file3 = file_base + '3.csv'
destination3 = destination_base + '3_es.csv'

file4 = file_base + '4.csv'
destination4 = destination_base + '4_es.csv'

file5 = file_base + '5.csv'
destination5 = destination_base + '5_es.csv'
```

Fig. 28. Almacenamiento de las rutas y destinos de los archivos a traducir y traducidos.

La siguiente funcionalidad crea una colección vacía para almacenar en ella los hilos de procesamientos a usar, seguido de variables llamadas “progress” que se encargarán de mostrar el progreso de cada hilo durante la ejecución.

```
# Crea una lista para almacenar los hilos
threads = []
```

Fig. 29. Creación de lista para almacenar los hilos a usar.

Seguido, la función “execute_traduction” ejecuta otra función llamada “traductionEs”, que es la encargada de traducir al español, pero en esta parte se ejecuta dicha función en su hilo correspondiente, creando la ejecución en paralelo.

```
def execute_traduction(file, language, destination):
    thread = threading.Thread(target=traductionEs, args=(file, language, destination))
    thread.start()
    threads.append(thread)

# Llama a las funciones en hilos separados
execute_traduction(file1, 'es', destination1)
execute_traduction(file2, 'es', destination2)
execute_traduction(file3, 'es', destination3)
execute_traduction(file4, 'es', destination4)
execute_traduction(file5, 'es', destination5)

# Espera a que todos los hilos terminen
for thread in threads:
    thread.join()
```

Fig. 30. Función que crea un hilo de procesamiento.

La función de traducción se ejecuta por cada hilo de procesamiento, en esta ocasión se recorrerá cada fila por la columna llamada “Message” del conjunto de datos, traduciendo cada una de ellas, para este caso se están manejando los errores ya que si llegase a ocurrir uno esto evita que toda la ejecución se detenga, sino que intente con el siguiente valor, además se muestra por pantalla el progreso actual de la ejecución para poder saber qué fila de qué columna y de qué hilo está traduciendo.

```python
import pandas as pd
from googletrans import Translator
import threading

def traductionEs(file, languaje, destination):
    try:
        translator = Translator()
        df = pd.read_csv(file)
        for i in range(len(df['Message'])):
            try:
                message = df['Message'][i]
                translated_message = translator.translate(message, dest=languaje).text
                df['Message'][i] = translated_message
                df.to_csv(destination, index=False)
                print('Hilo:', threading.current_thread().name, '- Guardado:', i+1, '/', len(df['Message']))
            except:
                print('Hilo:', threading.current_thread().name, '- Error:', i+1, '/', len(df['Message']))
    except Exception as e:
        print(e)
        print('Error al leer el archivo')
```

Fig. 31. Función de traducción de cada parte del conjunto de datos en su respectivo hilo.

La salida de información de la ejecución de la función de la Figura 30, es la información actual del procesamiento de datos para la traducción del contenido, en donde cada hilo, en este caso representado por “Hilo: Thread-n°” es representada por un hilo distinto, en donde dentro de ellos se está procesando una parte del conjunto de datos

```
Hilo: Thread-1 (traductionEs) - Guardado: 316 / 5956
Hilo: Thread-5 (traductionEs) - Guardado: 266 / 5955
Hilo: Thread-4 (traductionEs) - Guardado: 262 / 5956
Hilo: Thread-2 (traductionEs) - Guardado: 250 / 5956
Hilo: Thread-3 (traductionEs) - Guardado: 258 / 5956
Hilo: Thread-1 (traductionEs) - Guardado: 317 / 5956
Hilo: Thread-4 (traductionEs) - Guardado: 263 / 5956
Hilo: Thread-5 (traductionEs) - Guardado: 267 / 5955
Hilo: Thread-2 (traductionEs) - Guardado: 251 / 5956
Hilo: Thread-3 (traductionEs) - Guardado: 259 / 5956
Hilo: Thread-1 (traductionEs) - Guardado: 318 / 5956
Hilo: Thread-4 (traductionEs) - Guardado: 264 / 5956
Hilo: Thread-5 (traductionEs) - Guardado: 268 / 5955
```

Fig. 32. Traducción del conjunto de datos enron separados por cada hilo de procesamiento.

Como resultado final de la ejecución del procesamiento en hilos, se obtuvo las partes del conjunto de datos traducida.

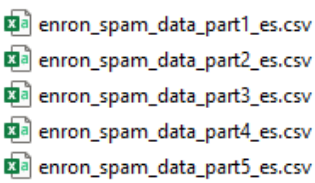


Fig. 33. Partes del conjunto de datos traducida.

2.5.1.3.Transformación/reducción.

Combinación de conjunto de datos

En la fase anterior durante la etapa de traducción del conjunto de datos a español por medios de hilos, se dividió el conjunto de datos Enron en cinco subconjuntos de datos iguales, en donde como resultado de esa implementación se pudo obtener una traducción de manera más eficiente y ahorrando tiempo.

En esta etapa se procederá a fusionar cada parte en un único conjunto que incluya todos los textos traducidos al español, permitiendo tener un conjunto consolidado al momento de entrenar al algoritmo.

Para lograr esto, se ejecutó el siguiente código donde se almacena en variables las rutas de los subconjuntos de datos, seguido se emplea una función de las librerías "pandas" para poder concatenar dichas partes y devolver un solo archivo.

```
import pandas as pd
path_base = '                                    SPAM_DETECTED/pre-datasets2/traduction2/prueba/'

file1 = path_base + 'enron_spam_data_part1_es.csv'
file2 = path_base + 'enron_spam_data_part2_es.csv'
file3 = path_base + 'enron_spam_data_part3_es.csv'
file4 = path_base + 'enron_spam_data_part4_es.csv'
file5 = path_base + 'enron_spam_data_part5_es.csv'

df1 = pd.read_csv(file1)
df2 = pd.read_csv(file2)
df3 = pd.read_csv(file3)
df4 = pd.read_csv(file4)
df5 = pd.read_csv(file5)

merged_df = pd.concat([df1, df2, df3, df4, df5])
merged_df.to_csv(path_base + 'enron_spam_data_es.csv', index=False)
```

Fig. 34. Código para fusionar los subconjuntos de datos a uno solo.

La salida de información de la ejecución del código previo es la combinación de todas las partes anteriormente divididas para poder traducir cada subconjunto en el idioma español de manera más efectiva, de esta forma se puede obtener un procesamiento más rápido respecto a lo planteado en la Tabla 12.

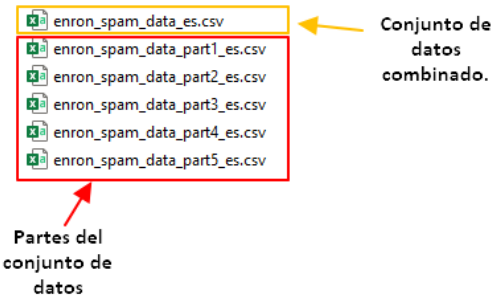


Fig. 35. Resultado final de la combinación de los subconjuntos de datos.

Siguiendo los mismos pasos, se procedió a combinar el conjunto de datos fraud_email y Enron, obteniendo un solo conjunto de datos, pero antes de eso se procedió a establecer valores predefinidos para el conjunto fraud_email, ya que el spam lo definía como 1 mientras que el ham como 0.

text	label
Calidad de suministro Dimensiones E...	spam
encima. SidAvísame. Gracias.	ham
Estimado amigo, Saludos a usted. De...	spam
SEÑOR. CHEUNG PUIHANG SENG BA...	spam
No es una evaluación sorprendente...	ham

Fig. 36. Resultado del cambio de valores en la columna label del conjunto de datos fraud_email.

Como resultado final, se procedió a combinar los conjuntos de datos para poder entrenar a los algoritmos seleccionados en la siguiente fase y poder analizar sus resultados para posteriormente poder seleccionar uno solo.

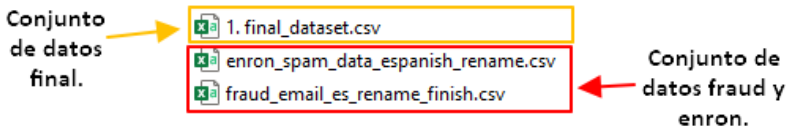


Fig. 37. Conjuntos de datos fraud, enron y combinación de ambos.

Tokenización

Al aplicar la técnica de tokenización se obtendrá los campos de los textos del conjunto de datos separadas en unidades más pequeñas llamadas tokens, en este caso se separa el texto en cada palabra que este contenga, permitiendo que

descomponer el texto y poder realizar un mejor análisis en una etapa posterior, teniendo como ventaja la reducción de los textos completos.

Text	Class
['Calidad', 'de', 'suministro', 'Dimensiones', 'EXCLUSIVAS', 'de', 'China', 'a', 'un', 'precio', 'inigualable', '.', 'Esti...	1
['encima', '.', 'SidAvísame', '.', 'Gracias', '.']	0
['Estimado', 'amigo', ',', 'Saludos', 'a', 'usted', '.', 'Deseo', 'abordarlo', 'con', 'una', 'solicitud', 'que', 'sería', 'd...	1
['SEÑOR', '.', 'CHEUNG', 'PUIHANG', 'SENG', 'BANK', 'LTD.DES', 'VOEUX', 'RD', '.', 'SUCURSAL', ',', 'HONG', 'KO...	1
['No', 'es', 'una', 'evaluación', 'sorprendente', 'de', 'la', 'Embajada', '.']	0
['Monica', '-Huma', 'Abedin', '<', 'Huma', '@', 'clintonemail.com', '>', 'Martes', ',', '29', 'de', 'junio', 'de', '2010...	0
['Pis', 'print.H', '<', 'hrod17', '@', 'clintonemail.com', '>', 'Jueves', '8', 'de', 'octubre', 'de', '2009', '8:01', 'PM'Jil...	0
['Estimado', 'Tom', ':', 'H', '<', 'hrod17', '@', 'clintonemail.com', '>', 'Viernes', ',', '11', 'de', 'diciembre', 'de', '2...	0
['Saludos', 'del', 'abogado', 'Robert', 'Williams=2CEstimado', 'amigo=2C', 'Sé', 'que', 'mi', 'carta', 'le', 'llegará...	1

Fig. 38. Resultado de aplicar tokenización al conjunto de datos.

2.5.1.4.Minería de datos.

En esta etapa se seleccionaron tres algoritmos de aprendizaje automático, basándonos en el estudio "Clasificación del spam mediante aprendizaje automático y procesamiento del lenguaje natural" [95], dónde los algoritmos con mejor promedio de precisión fueron "Support Vector Machine", "Random Forest" y "Naive Bayes (Multinomial)" cada uno superando el 95% de rendimiento, aunque esto es independiente del conjunto de datos seleccionado, ya que en el trabajo titulado como "Diseño E Implementación De Una Extensión Chrome Para La Detección De Sitios Web De Phishing Utilizando Aprendizaje Automático" [64] se aplicaron dos de los mismos algoritmos pero con un distinto conjunto de datos, donde el algoritmo "Random Forest" tuvo una precisión de 96%-98%, mientras que el algoritmo Naive Bayes obtuvo un rendimiento de 83% - 92.5% aproximadamente.

Los algoritmos seleccionados en esta fase para posteriormente entrenarlos son:

- Random forest
- Decision Tree
- Naïve Bayes

Random Forest

Entrenamiento

Para poder entrenar y obtener los resultados del mismo para este algoritmo, se procedió a usar la librería scikit learn que proporciona diversidad de algoritmos

dedicados al aprendizaje automático, además de herramientas que permiten calcular las precisiones de los entrenamientos por medio de métricas de evaluación.

El siguiente código importa las librerías a usar para poder entrenar el modelo, seguido de una variable denominada "chunk_size = 10000", se estableció este valor porque el total de filas es de 40027, asignando este valor se ejecutará esa cantidad por cada iteración, acelerando el procesamiento de datos ya que no almacena el conjunto de datos en memoria. Se crean variables para el entrenamiento y pruebas respectivas, especificando en la variable "test_size" que se usará el 20% del conjunto total de los datos para el entrenamiento respectivo.

Se crea un modelo donde se almacena en el un objeto de clasificación del algoritmo, donde por parámetro se establece el valor de 100, creando esta cantidad de árboles de decisión, permitiendo tener una mayor precisión, pero a su vez, el tiempo empleado para poder entrenar y evaluar este modelo es mucho mayor.

Por último, se presentan por consola los resultados del entrenamiento por medio de la variable Accuracy y report, donde el primero es una estimación de las predicciones correctas respecto al total de predicciones realizadas, mientras que el report muestra más métricas de evaluación demostradas en las tablas 13 y 14.

En el Anexo 7 se establece el proceso que se siguió para poder entrenar el algoritmo Random Forest con el conjunto de datos obtenido en las fases anteriores.

Resultados

Durante el entrenamiento usando el conjunto de datos no tokenizado, se observó que el proceso tardó alrededor de 3 minutos, usándose 32021 datos se entrenamiento, mientras los restantes de pruebas.

Tabla 13 - Resultados del entrenamiento del algoritmo Random Forest con el conjunto de datos sin tokenizar.

	Precision	**Recall**	**F1-score**	**Support**
ham	0.97	0.98	0.98	4384
spam	0.97	0.97	0.97	3662
accuracy			0.9731	8006

Durante el proceso de entrenamiento del algoritmo, pero esta vez con el conjunto de datos tokenizado, se pudo apreciar que el proceso tardó aproximadamente 10 minutos, tanto entrenamiento como pruebas, esto se debe a que el conjunto de datos tokenizado separa un texto completo en palabras, lo cual llega hacer el conjunto de datos mucho más robusto, sin embargo, los resultados fueron inferiores frente al entrenamiento con el conjunto de datos no tokenizado.

Tabla 14 - Resultados del entrenamiento del algoritmo Random Forest con el conjunto de datos tokenizado.

	Precision	**Recall**	**F1-score**	**Support**
ham	0.95	0.99	0.97	4415
spam	0.99	0.94	0.96	3591
accuracy			0.9665	8006

Decision Tree

Entrenamiento

El código para poder realizar el entrenamiento para este algoritmo es similar al de los algoritmos random forest y naive bayes, ya que provienen de la misma librería, pero tienen pequeñas diferencias, en este caso la importación para poder crear una instancia del algoritmo cambia, en este caso importándose de otro módulo.

Para esta ocasión en las variables de entrenamiento y pruebas, no se establece un número que específique la cantidad de árboles de decisiones que se usarán, ya que por defecto el valor será de 100. Para poder entrenar el algoritmo tanto con el conjunto de datos sin tokenizar y no tokenizado, solo hay que cambiar la ruta del conjunto de datos.

En el Anexo 8 se establece el proceso para poder ejecutar el entrenamiento del algoritmo Decision Tree.

Resultados

Durante el tiempo de ejecución del algoritmo con el conjunto de datos sin tokenizar, se apreció que el tiempo de ejecución fue de aproximadamente 5 minutos entre los datos de entrenamiento y pruebas, teniendo los siguientes datos, en la que se usó un total de 8006 del conjunto de datos original, equivaliendo a un 20% tanto en datos divido entre spam y ham.

Tabla 15 - Resultados del entrenamiento del algoritmo Decision Tree con el conjunto de datos sin tokenizar.

	Precision	**Recall**	**F1-score**	**Support**
ham	0.94	0.93	0.94	4384
spam	0.92	0.93	0.93	3622
Accuracy			0.9323	8006

El entrenamiento del algoritmo con el conjunto de datos tokenizado se aproximó a 10 minutos de espera, en donde los resultados mejoraron en cuanto a las métricas de evaluación de Recall y F1-score respecto al ham, pero se vieron disminuidas frente las métricas de precisión, Recall y f1-score, teniendo como resultado total una disminución aproximada de 1%.

Tabla 16 - Resultados del entrenamiento del algoritmo Decision Tree con el conjunto de datos tokenizado.

	Precision	**Recall**	**F1-score**	**Support**
ham	0.92	0.94	0.93	4415
spam	0.93	0.90	0.91	3591
Accuracy			0.9245	8006

Naïve Bayes

Entrenamiento

Para poder hacer uso del algoritmo de naive bayes, fue importado desde una función establecida por scikit learn, a diferencia de los algoritmos de Random Forest y Decision Tree, no se establecen múltiples de subramas para poder tomar una decisión, sino que se basa en el teorema de Bayes y asume que todas las

características que entrena y avalúa son independientes entre sí, enfocándose en cálculos de probabilidad.

Asimismo, se establecen como parámetro que se seleccionarán solo el 20% del subconjunto total para poder realizar las pruebas, mientras que el resto será de entrenamiento.

En el Anexo 9 se establece el proceso para poder entrenar el algoritmo Naive Bayes.

Resultados

Los resultados obtenidos en el entrenamiento con los datos sin tokenizar se obtuvieron resultados en la mayoría de las métricas de 0.94, equivalente al 94% e incluso llegando hasta el 99%, dando como un resultado primero del 0.96, equivalente al 96%

Tabla 17 - Resultados del entrenamiento del algoritmo Naïve Bayes con el conjunto de datos sin tokenizar.

	Precision	**Recall**	**F1-score**	**Support**
ham	0.99	0.95	0.97	4384
spam	0.94	0.99	0.96	3622
accuracy			0.9660	8006

Al momento de establecer el entrenamiento con el conjunto de datos tokenizado, se puedo apreciar que se redujeron el rendimiento respecto al otro conjunto de datos, donde las métricas más afectadas fueron la Precision respecto al spam, el Recall en la categoría de ham y una reducción en el F1-score en ambas categorías.

Tabla 18 - Resultados del entrenamiento del algoritmo Naïve Bayes con el conjunto de datos tokenizado.

	Precision	**Recall**	**F1-score**	**Support**
ham	0.99	0.91	0.95	4384
spam	0.90	0.99	0.94	3622
accuracy			0.9454	8006

2.5.1.5.Interpretación/evaluación.

Durante el proceso de las etapas anteriores, se realizó la selección de conjuntos de datos para poder procesarlos y realizar una preparación de estos de acuerdo a

parámetros establecidos, tales como eliminar aquellas características que no tendrían relevancia al momento de entrenar los algoritmos y aquellos que no cuentan con el formato adecuado.

Para limpiar y procesar los datos se implementaron diversas técnicas como la traducción al idioma español, eliminación de valores duplicados y nulos, además de realizar un conteo de los valores totales del conjunto de datos Enron, determinando que la cantidad era relativamente grande por ello se aplicó la técnica de separar el conjunto de datos en segmentos pequeños y aplicar hilos de procesamiento para realizar la traducción de manera más rápida, para posteriormente estas ser unidas en el conjunto de datos original.

Después del procesamiento y limpieza de datos se procedió aplicar la técnica de tokenización al conjunto de datos para poder realizar un entrenamiento de manera más eficiente, donde para aplicar esto se seleccionaros tres algoritmos de aprendizaje automático para su posterior análisis.

Durante la fase de minería de datos se puso a prueba a efectividad de cada algoritmo con el conjunto de datos que contenía los datos tokenizados y sin tokenizar, dando diferentes resultados como se muestran en dicha fase, dando como al algoritmo de Random Forest con las mejores estadísticas en cuento a los entrenamientos del mismo conjunto de datos, pero con técnicas diferentes.

El algoritmo por emplear en este proyecto será el de Random Forest debido a las estadísticas obtenidas durante las diversas fases aplicadas en la metodología KDD, obteniendo resultados superiores frente a algoritmos como Naïve Bayes y Decision Tree.

2.5.2. Metodología OMSTD

2.5.2.1.Estructura del agente detector de spam

2.5.2.1.1. Selección de lenguaje de programación

Existen diversos lenguajes de programación que han ganado popularidad durante los últimos 20 años tales como JavaScript, Java, Python, etc., en donde cada lenguaje tiene su propio propósito pese a que se pueden crear las mismas herramientas con diferentes lenguajes de programación [99], por ello que se realiza

la siguiente comparación entre estos lenguajes de programación para conocer sus debilidades y fortalezas, para poder seleccionar uno solo que servirá como base en el desarrollo de la propuesta.

Tabla 19 - Comparativa entre JavaScript, Java y Python [99], [15], [100], [101]

	JavaScript	**Java**	**Python**
Tipo	Interpretado	Compilado	Interpretado
Propósito	• Apps móviles • Apps de escritorio • Backend (NodeJS) • Frontend	• Aprendizaje automático • Apps móviles • Apps de escritorio • Backend • Frontend	• Aprendizaje automático • Scripts de automatización • Apps de escritorio • Backend • Frontend • Apps de scraping
Paradigma	Orientado a prototipos	Orientado a objetos	Multiparadigma
Portabilidad	Media	Baja	Alta
Librerías	Alta	Alta	Alta
Curva de aprendizaje	Alta	Media	Alta
Dependencia	Web	JDK	Intérprete de Python
Tipado	Dinámico	Estático	Dinámico

La amplia variedad de librerías disponibles en Python lo convierten en una de las mejores opciones para el desarrollo de scripts, automatización y otros productos relacionados con la ciberseguridad. Además, la sintaxis limpia y fácil de entender de Python facilita el desarrollo rápido de programas [102].

Por lo tanto, el uso de Python en el diseño y desarrollo es altamente recomendado en la investigación actual.

2.5.2.1.2. Selección de tipo base de datos

Seleccionar el tipo de base de datos y un gestor apropiado, involucra tener en cuenta diversos factores relacionados con el proyecto y la adaptabilidad, en este caso la base de datos a seleccionar deberá servir como base para el desarrollo de otros componentes del proyecto y que este no tenga inconvenientes al momento de su uso e implicar seleccionar otro tipo de base de datos.

Tabla 20 - Comparación entre base de datos SQL Vs NoSQL [103], [104], [105]

Característica	**SQL**	**NoSQL**
Estructura	• Relacional • Tablas	• Columnas • Grafos • Clave-valor • Documento
Escalabilidad	Vertical	Horizontal
Esquema	Esquemas dinámicos	Esquemas predefinidos
Consistencia y disponibilidad	Alta	Alta
Software	SQL Server, Oracle, MySQL, etc.	DynamoDB, MongoDB, Cassandra, etc.
Mapeo de objetos de datos	Requiere ORM (Mapeo Relacional De Objetos)	No requiere

Para el proyecto se seleccionará el tipo de base de datos NoSQL, ya que permitirá tener una flexibilidad en el esquema de datos, ya que trabajar con la detección de correos electrónicos como spam implica trabajar con datos estructurados, semiestructurados y no estructurados, implicando no tener un esquema definido durante el desarrollo y cambiando conforme sea necesario, involucrando una

escalabilidad horizontal y consultas como operaciones de lecturas de manera más eficiente.

2.5.2.1.3. Selección de Sistema de gestión de base de datos (SGBD).

Una vez seleccionado el tipo de base de datos, se tendrá que seleccionar adecuadamente el SGBD para el proyecto, teniendo en cuenta las funcionalidades que ofrecen, escalabilidad y rendimiento, soporte y el costo, realizando una comparación entre diferentes SGBD de NoSQL, teniendo en cuenta sus diferentes fortalezas y debilidades comparado a los requisitos del proyecto.

Tabla 21 - Comparación entre Firestore, MongoDB y CassandraDB [106], [107], [108], [109]

Característica	**Firestore**	**MongoDB**	**CassandraDB**
Tipo de base de datos	• Documento • Clave-valor	• Documento • Clave-valor	Columna amplia
Servicio	Nube	• Nube • Local	Local
Escalabilidad	Alta	Alta	Alta
ACID (Atomicidad, consistencia, aislamiento y durabilidad)	Soporta	Soporta	No soporta
Código abierto	No	Sí	Sí
Casos de uso	• Apps móviles • Apps webs • Desarrollo en servidores • Análisis en tiempo real	• Minería de datos • Internet de las cosas • Almacenamiento y registro de eventos • Desarrollo en general	• Aprendizaje profundo • Minería de datos • Sistemas bancarios • Publicidad

Característica	Firestore	MongoDB	CassandraDB
Soporte para Python	Sí	Sí	Sí

Se seleccionó MongoDB para el desarrollo del proyecto por la combinación de características documental y clave-valor brindando una gran flexibilidad al momento de adaptarse a las necesidades cambiantes del proyecto, además de soportar el desarrollo general de aplicaciones, brindando gran variedad de casos de usos.

Otro factor importante fue el soporte que tiene con el lenguaje Python, que fue seleccionado previamente para el desarrollo del proyecto, facilitando la integración con la base de datos de una manera sencilla.

2.5.2.1.4. Organización y estructura

Durante esta fase se implementó la programación estructural para poder desarrollar el script que detecte spam, de esta forma se separó el código en diferentes carpetas denominadas módulos y asignar una única responsabilidad a cada una de ellas, permitiendo un escalamiento horizontal.

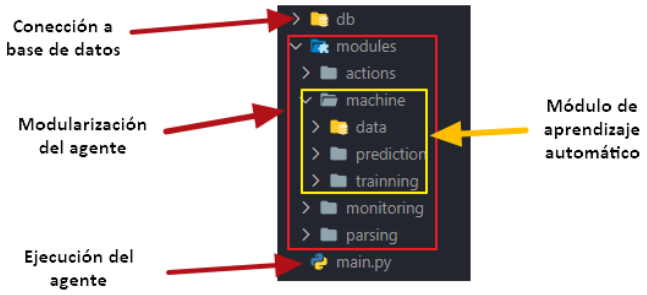


Fig. 39. Estructura del agente aplicando programación estructural.

La base de datos MongoDB permite crear colecciones para almacenar la información, para el agente detector de spam, se harán uso de las siguientes colecciones:

La siguiente colección llamada “alert_sent_user”, hace referencia al registro de la alerta enviada al usuario, en ella no se almacena ninguna información confidencial

del correo electrónico, solo la hora en que se envió la alerta al mismo, esto para poder tener una estadística de las cantidades de alertas que el agente envía.

Fig. 40. Colección alert_sent_user.

La siguiente colección llamada “normal_mail”, hace referencia al tiempo en el que el agente detectó un correo normal y este insertó la fecha del mismo con la predicción, en este caso tampoco se insertan datos personales.

Fig. 41. Colección normal_mail.

2.5.2.1.5. Entrada y salida de información

Entrada de información indirecta:

- Carpeta de Zimbra que será analizada para la detección de nuevos correos electrónicos, caso contrario se establecerá la ruta de la carpeta por defecto.
- Ingreso del correo electrónico de la cuenta que enviará las alertas a los usuarios.
- Ingreso de claves secretas para generar los tokens.

Salida de información:

- Correo electrónico detectado como spam.
- Mensajes mostrados en la consola sobre el análisis del nuevo correo electrónico.
- Almacenamiento de datos la hora del análisis de los correos como ham y spam.
- Envío de alerta a través de un mensaje de correo electrónico.
- Mensajes por consola de errores.

2.5.2.1.6. Redistribuciones

El agente estará solo disponible para versiones de Zimbra en distribuciones de Linux, puesto que este servicio solo está disponible en este sistema operativo, desarrollado en un script de Python siguiendo programación modular.

La siguiente función recibe como parámetro la ruta del correo electrónico detectado como nuevo para poder ser preprocesado, en este caso se filtrará a dicho correo por la palabra “incoming”, queriendo decir que es un correo electrónico que aún no se ha enviado, sino que es un borrador.

```
def get_non_ignored_directory(path):
    try:
        words = ['incoming']
        if any(word in path for word in words):
            return False
        return True

    except Exception as e:
        prin('No tiene permisos para leer el directorio' + str(e))
```

Fig. 42. Función que filtra por nombre de la ruta a analizar.

Una vez preprocesado el nombre de la ruta y pasar el primer filtro, se procederá a analizar de vuelta el correo, pero en esta ocasión abriendo el archivo para analizar su contenido, en este caso también se validará que el correo haya sido recibido por el usuario con la palabra “Received”, para esta ocasión se analizará línea por línea en correo electrónico para así cuando encuentre la coincidencia retorne el valor, así no analiza todo el correo innecesariamente.

```
def get_file_with_received(path):
    try:
        with open(path, 'r') as file:
            valids_words = ['Received']
            for line in file:
                if any(word in line for word in valids_words):
                    file.close()
                    return True
            file.close()
            return False
    except Exception as e:
        print('Ha ocurrido un error en la función get_file_with_received' + str(e))
```

Fig. 43. Función que filtra características de correos electrónicos para excluir del análisis.

Realizado los filtros para asegurarse que un correo electrónico no es un borrador y que ha sido recibido por el remitente, se procederá a ejecutar la siguiente funcionalidad para poder tener identificado al correo electrónico que procederá a

analizar, en este caso se recibirá la ruta procesada y ésta será desestructurada, tal que la función retorne el identificador del correo electrónico y el nombre del archivo, de esta forma cuando se quiera alertar al administrador de posibles spam éste cuente con estos datos esenciales para poder realizar acciones sobre él.

```python
def get_name_file_and_id(path):
    try:
        path = path.strip()
        file = path.rsplit('/', 1)[-1]
        id = file[:file.find('-')]
        return id, file
    except:
        print('la ruta del archivo no cumple con el formato de zimbra')
```

Fig. 44. Función que extrae el id del correo electrónico junto con el nombre del archivo del mensaje.

La siguiente función es empleada para extraer las partes esenciales del correo electrónico, como el remitente, receptor, fecha, mensaje adjunto y el mensaje, donde esto es guardado en un diccionario para luego ser retornado y ser usado en cualquier lado del agente, por ejemplo, para mostrarlo por consola, predecir si un correo es spam, insertar en base de datos o alertar al administrador con los detalles del correo.

```python
def read_email_content(path):
    try:
        with open(path, "rb") as f:
            # Crea un diccionario vacío
            email = {}
            msg = mailparser.parse_from_bytes(f.read())
            email['date'] = msg.headers.get('Date', 'Fecha no encontrada')
            email['from'] = msg.from_[0][1]
            email['to'] = [x[1] for x in msg.to]
            email['id_group_mail'] = msg.headers.get('Message-ID', 'Id no encontrado')
            email['subject'] = msg.headers.get('Subject', 'Asunto no encontrado')

            body = msg.text_plain[0] if len(msg.text_plain) > 0 else 'No hay mensaje'
            body = delete_spaces_between_lines(body)
            email['body'] = body

            email['attachments'] = []
            for file in msg.attachments:
                email['attachments'].append(file.get('filename'))

            f.close()
            return email
    except Exception as e:
        print('Ha ocurrido un error en la función read_email_content' + str(e))
```

Fig. 45. Función que extrae las características de un correo electrónico previo a una ruta a leer.

Para poder tener una estadística de los correos electrónicos detectados como spam, se procederá a usar la siguiente función, donde como parámetros recibidos son algunas características extraídas por la función anterior,

```python
def insert_ham(prediction):
    try:
        collection = db["normal_mail"]
        verify_insert = collection.insert_one(
            {
                'dateOfAnalysis': datetime.now(),
                'prediction': prediction

            })
        if not verify_insert:
            return False
        return True
    except:
        print('Error al insertar los datos')
```

Fig. 46. Función que almacena en la base de datos información del correo electrónico detectado como spam.

Se implementará la siguiente función para alertar al usuario de que un correo que llegó a su cuenta ha sido detectado como spam y removido a la carpeta perteneciente, seguido de la opción para que este puede reportar el usuario detectado, caso contrario lo puede pasar por desapercibido.

```python
html_body = f"""
            <html>
                <body>
                    <p><b>**Este mensaje es generado automáticamente por un detector de spam, por favor no responda a este correo**</b></p>
                    <p><b> El correo ha sido redireccionado a la carpeta de spam.</b></p>
                    <p><b> Si considera que este correo no es spam, ignore este mensaje.</b></p>
                    <p>Los detalles del correo son los siguientes:</p>
                    <p>Desde: {from_}</p>
                    <p>Para: {to}</p>
                    <p><b>Asunto:</b> {subject}</p>
                    <p><b>Cuerpo:</b> {body}</p>
                    <p>Si desea reportar este correo, por favor haga clic en el siguiente botón para más detalles:</p>
                    <p>
                        <a href="http://192.168.23.47:5000/quarantine/validate/{modified_token}">
                        <button style="padding: 10px 20px; background-color: #800020; color: white; border: none; border-radius: 4px; cursor: pointer;">
                        Reportar</button></a>
                    </p>
                </body>
            </html>
        """
```

Fig. 47. Función que envía una alerta a través de Zimbra al usuario.

2.5.2.1.3. Despliegue

Para asegurar la efectividad y funcionalidad del algoritmo desarrollado, se lleva a cabo su despliegue en un entorno controlado. En esta etapa, se implementa el algoritmo en un entorno específicamente preparado para su evaluación y puesta en fase de desarrollo, permitiendo realizar una simulación de un entorno real de producción, ajustando y optimizando el algoritmo conforme se realicen las pruebas.

La siguiente figura ilustra el envío de un correo electrónico que contiene información declarada como spam, se realizará esta prueba para ver la eficacia del agente para detectar este tipo de correos.

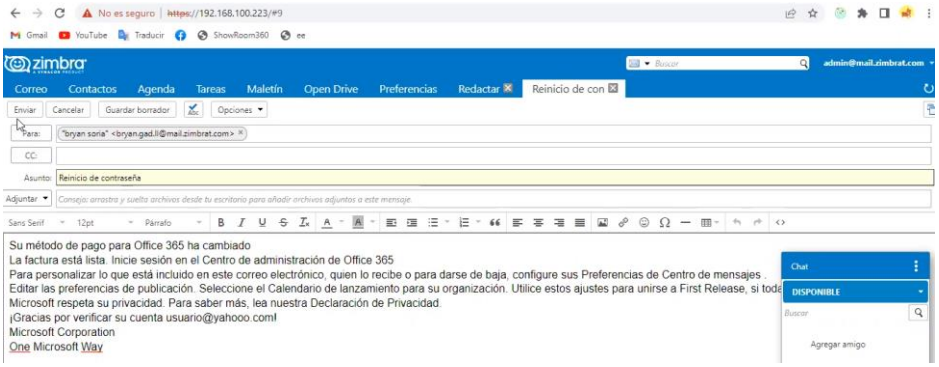


Fig. 48. Envío de mensajes con contenido spam mediante un entorno controlado.

Enviado el correo electrónico y aplicado los filtros previos para considerar un correo electrónico válido, el agente resalta las características del correo electrónico, detectado como spam.

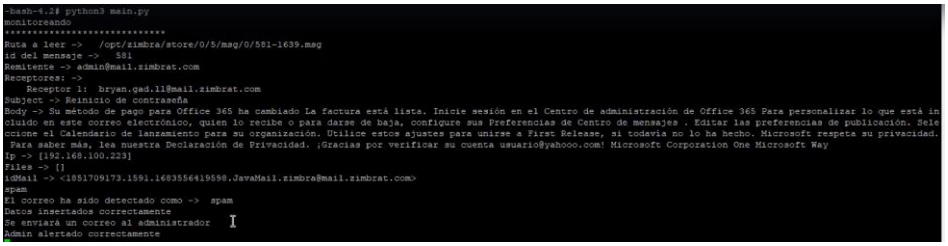


Fig. 49. Detección de spam mediante el agente en un entorno controlado.

Cuando el agente detecte un correo electrónico como spam, este será removido desde la bandea de entrada del usuario hasta la carpeta de spam del usuario, si alterar el mensaje solamente cambiándolo de ubicación.

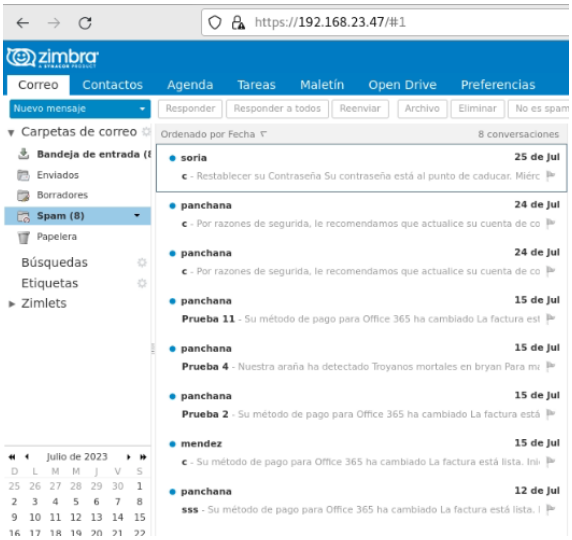


Fig. 50. Correos electrónicos categorizados como no deseados removidos hacia la carpeta de spam por el agente en un entorno controlado.

El usuario recibe la alerta por parte del correo electrónico “detected”, dónde le resalta características como remitente, receptor, asunto, mensaje, la ruta original y la ruta donde se puede leer el mensaje completo para analizarlo de mejor manera.

Donde primero se avisa al usuario que el correo ha sido generado automáticamente por el agente detector de spam, seguido de los datos importantes del correo electrónico detectado, tales como el remitente y el receptor, en este caso así el correo cuente con más de un receptor, se le enviará solo el mensaje de alerta al usuario correspondiente, de esta manera si un usuario reporta que lo haga solo para él y no para los otros receptores del correo.

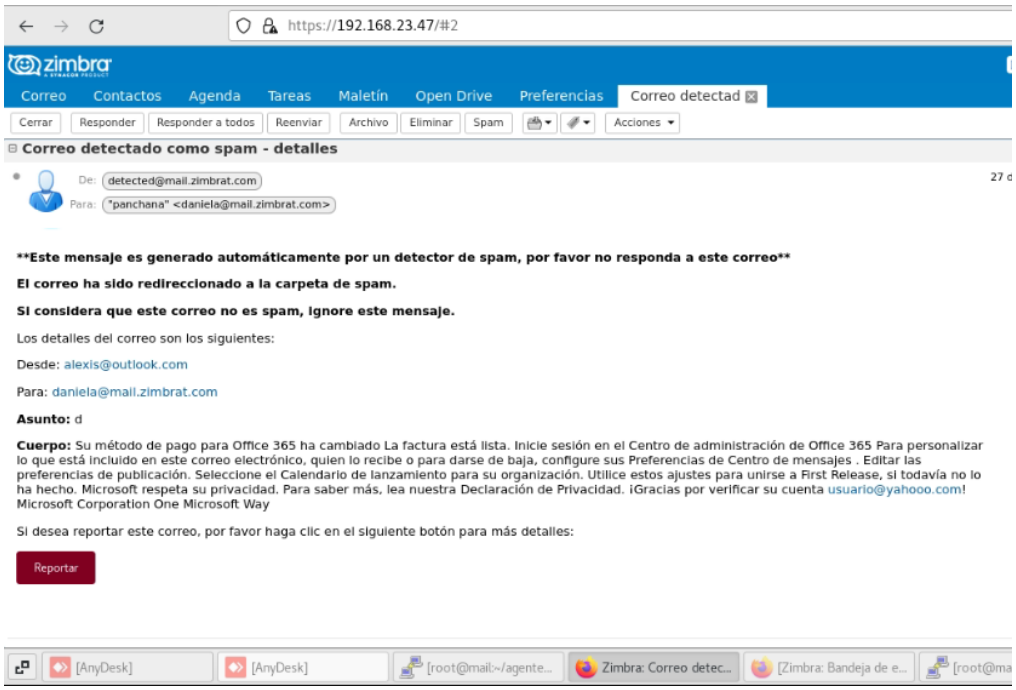


Fig. 51. Alerta enviada por el agente y recibida por el usuario.

2.5.2.2.Estructura del backend

2.5.2.2.1. Selección de framework para el backend orientado a Python

Para poder seleccionar un marco de trabajo que se pueda integrar junto con nuestro agente y el frontend, se procederá a evaluar varias características claves entre flask, django y fastapi desde la perspectiva del proyecto y la escalabilidad que cada uno de ellos pueda ofrecer.

Tabla 22 - Comparación entre frameworks backend con Python [110], [111], [112]

Característica	Flask	Django	FastAPI
Tipo	Micro-framework	Framework	Micro-framework
Integración con API RESTs	Ampliamente usado	Ampliamente usado	Especialmente diseñado
Curva de aprendizaje	Baja	Media	Media
Flexibilidad	Alta	Alta	Alta
Plantillas	Plantilla jinja2	Plantilla django	No posee
Integración con base de datos	Amplia variedad de base de datos	Múltiple variedad de base de datos	Múltiple variedad de base de datos

Luego de la comparativa entre las diversas características de los framewoks, se concluyó que flask se adapta de mejor manera al proyecto, ya que ofrece una buena integración con API RESTs, además su baja curva de aprendizaje fue la que resultó por encima de frameworks como django y fastapi.

2.5.2.2.2. Organización y estructura

Para desarrollar el backend se usó el framework Flask de Python para el desarrollo web, flask no impone una estructura que se deba seguir para poder crear aplicaciones en él, pero si da recomendaciones para seguir una mejor organización de carpetas y se optó por la siguiente estructura:

db: contiene archivos relacionados con la base de datos, como modelos de datos y configuraciones de conexión, etc., manejándose todas las operaciones relacionadas con la base de datos.

logic: contiene archivos relacionados con la ejecución de comandos de Zimbra, que son ejecutados cuando se realiza una petición hacia una API rest, separando la lógica de las rutas y ejecución de comandos del sistema.

middleware: son componentes que se ejecutan antes o después de que una solicitud llegue al controlador, y se utilizan para realizar tareas como autenticación, autorización, manipulación de solicitudes y respuestas.

routes: se encuentran archivos que definen las rutas y los controladores de la aplicación. Cada archivo contiene las rutas relacionadas con un determinado conjunto de funcionalidades y sus respectivos controladores. Aquí se manejan las solicitudes HTTP, se procesarían los datos y se enviarían las respuestas correspondientes al cliente.

app.py: contiene la entrada principal de la aplicación Flask.

main.py: contiene el código principal para iniciar la aplicación de Flask.

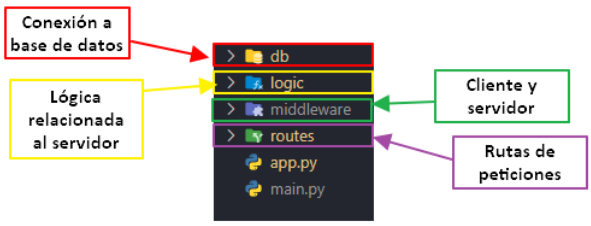


Fig. 52. Estructura del backend siguiendo el modelo de Flask.

Las colecciones que se harán uso en el backend serán las siguientes:

La colección reports hace referencia al correo electrónico reportado por el usuario mediante la alerta enviada por el agente, esta información solo se almacena cuando el usuario ha dado el consentimiento de poder hacer uso de este.

```
{
  _id: ObjectId("64b37705ee84b1241e99b725"),
  uuid: 'ee137b1f-3540-448b-9944-60a9fa9556a9',
  dateOfReport: ISODate("2023-07-15T23:50:13.987Z"),
  from: 'bryan@mail.zimbrat.com',
  to: 'andres@mail.zimbrat.com',
  subject: 'Cambiar método de pago',
  message: 'Su método de pago para Office 365 ha cambiado La factura está lista. I…',
  id_mail: '431',
  name_attachments: [],
  prediction: 'spam',
  path: '/opt/zimbra/store/0/3/msg/0/431-1316.msg',
  file_name: '431-1316.msg'
},
```

Fig. 53. Colección reports.

La colección user permite almacenar a los usuarios registrados, en donde los nuevos registrados tienen el estado de la clave active en false, ya que deben de esperar a que el administrador pueda conceder el registro.

De igual forma el admin puede establecer el valor de reject en true si es que quiere rechazar a un usuario.

Fig. 54. Colección user.

2.5.2.2.3. Entrada y salida de información

Entrada de información:

- Credenciales de inicio de sesión enviada a través del cliente.
- Peticiones realizadas en el servidor por petición del cliente.
- Aceptación o rechazo por parte del usuario al reportar un correo electrónico

Salida de información:

- Respuestas en formato JSON enviadas al cliente.

Fig. 55. Respuesta en formato JSON.

2.5.2.2.4. Redistribuciones

El backend que contendrá las APIs Rest estará disponible como un servidor que reciba peticiones del cliente, esta arquitectura permitirá separar la lógica del negocio en dos partes: el frontend (cliente) y el backend (servidor).

```python
@dashboard.route('/get-whitelist', methods=['GET'])
@token_required
def get_user_filter():
    try:
        blacklist = database['user_filter_list']
        type_user = request.json['type_user']
        mails = blacklist.find({'type_user': 'white'})
        mail_list = list(mails)
        for mail in mail_list:
            mail['_id'] = str(mail['_id'])
            mail['date'] = str(mail['date'])
        if len(mail_list) == 0:
            return jsonify({'ok': False, 'message': 'No hay correos', 'mails': []})
        response = {'ok': True, 'data':{
            'mails': mail_list,
        }}

        return Response(json_util.dumps(response), mimetype), 200
    except Exception as e:
        print(e)
        return jsonify({'message': 'Ocurrió un error al obtener los correos'}), 500
```

Fig. 56. Api Rest para obtener los correos electrónicos pertenecientes a la lista blanca.

2.5.2.2.5. Despliegue

El servidor estará disponible dentro del servidor de correo electrónico, para poder ejecutar los comandos que permitan remover un correo detectado como spam dentro del correo electrónico Zimbra, ya que se tendrá acceso directo a las funcionalidades y características propias de este servicio.

2.5.2.3.Estructura del frontend

2.5.2.3.1. Selección del framework web para el frontend

Al momento de seleccionar un framework de desarrollo, se deberá tomar en cuenta varios aspectos como la escalabilidad, soporte, arquitectura base y características específicas del marco de trabajo, esto con el fin de que cuando se implemente nuevas funcionalidades al dashboard, este sea capaz de adaptarse al crecimiento. Aunque la escalabilidad no esté totalmente ligada al framework o lenguaje de programación, estos ayudan a que esto sea posible de manera más fácil.

Tabla 23 - Comparación entre frameworks web de frontend [113], [114]

Características	**Angular**	**React**	**Vanilla JS**
Tipo	Framework	Librería	JavaScript puro
Lenguaje de programación	TypeScript	JavaScript	JavaScript
Soporte	Google	Facebook	Comunidad global
Arquitectura	Basado en MVC (Modelo-vista - controlador) o MVVC(Modelo-vista-vista-controlador)	No especifica arquitectura	No especifica arquitectura
Herramientas CLI (Interfaz de línea de comandos)	Cuenta con su propio CLI	No cuenta, aunque se puede instalar de terceros	No cuenta
Enlace de datos	Enlace bidireccional	Enlace unidireccional	No posee
Escalabilidad	Altamente escalable	Escalable	Escalabilidad limitada
Usos	Aplicaciones a gran escala	Aplicaciones donde se conoce el punto límite	Proyectos pequeños

Concluida la comparativa con puntos específicos que se espera para desarrollar el software, se llegó a la conclusión que angular se adapta al proyecto de manera sólida, ya que al ser un framework, este incluye diversas funcionalidades ya implementadas listas para ser usada, a diferencia de react que es una librería y para hacer uso de ciertas funcionalidades se debe instalar de terceros, dónde existen múltiples bibliotecas que hacen lo mismo, mientras que vanilla JavaScript es el

lenguaje de programación en estado puro, donde no es recomendado crear un proyecto relativamente grande.

2.5.2.3.2. Organización y estructura

Durante esta fase se implementó la estructura ya proporcionada por angular para desarrollar sistemas sobre este framework, la cual divide en varios puntos durante la estructura, que son los componentes, modelos, módulos, vistas, clases y servicios:

Componentes: representan parte de la interfaz de usuario y se encargan de gestionar su propia lógica, y se organizan jerárquicamente para construir la aplicación.

Modelos: representar los datos y su estructura, usados para definir el formato de los datos que se manejan en la aplicación permitiendo y facilitando la manipulación de datos.

Módulos: representan la agrupación de componentes, servicios y otros elementos relacionados con la funcionalidad, permitiendo organizar y modularizan la aplicación, aplicando la técnica de carga perezosa y mejorar el rendimiento.

Vistas: son plantillas HTML que definen la interfaz de usuario de la aplicación, usadas para mostrar los datos y permiten la interacción con el usuario

Clases: definen la lógica, comportamiento de los componentes y servicios, pueden contener métodos y propiedades que se utilizan para realizar tareas específicas.

Servicios: encapsulan la lógica y la funcionalidad compartida en la aplicación, proporcionando métodos y funciones que pueden ser utilizados por varios componentes para realizar tareas específicas, como obtener datos de un servidor, manejar la autenticación y realizar operaciones asíncronas.

Fig. 57. Estructura de la interfaz web aplicando la estructura del framework Angular.

2.5.2.3.3. Entrada y salida de información

Entrada de información:

- Registrar usuario
- Actualizar contraseña
- Credenciales del usuario.
- Selección del correo electrónico para ver los detalles.
- Añadir correo electrónico a lista negra
- Editar correo electrónico de la lista negra
- Eliminar correo electrónico de la lista negra
- Añadir correo electrónico a la lista blanca
- Editar correo electrónico de la lista blanca
- Eliminar correo electrónico de la lista blanca

Salida de información:

- Gráficas estadísticas sobre correos detectados como spam.
- Lista de correos electrónicos detectados como spam.
- Alertas de errores.
- Alertas de éxito.

- Archivo pdf con gráficas estadísticas
- Archivo xlsx con los datos sin procesarlos

2.5.2.3.4. Redistribuciones

El ambiente web o dashboard estará disponible como una redistribución en el servicio web, lo que permitirá a los usuarios acceder a él a través de Internet. Esto significa que los usuarios podrán ingresar servicio desde cualquier dispositivo con conexión a Internet, ya sea una computadora de escritorio, una laptop, una tableta o un dispositivo móvil.

La siguiente imagen muestra cómo serán manejadas las vistas del dashboard, siguiendo la estructura que proporciona angular.

```
<div class="flex flex-col mt-3 sm:flex-row w-full gap-4 justify-between sm:gap-8 p-4">
    <div class="bg-white rounded-xl p-2 shadow-lg flex w-full justify-between sm:h-1/2 sm:w-1/4">
        <div class="w-full">
            <h2 class="text-xs font-medium text-gray-600 mb-1">Correos Normales</h2>
            <p class="text-lg font-semibold text-gray-600 mb-1">{{cantHam}}</p>
        </div>
        <div class="w-1/4 flex justify-center items-center">
            <div class="bg-green-500 rounded-full w-12 h-12 flex items-center justify-center">
                <i class="fa-solid fa-user-check text-white text-2xl"></i>
            </div>
        </div>
    </div>

    <div class="bg-white rounded-xl p-2 shadow-lg flex w-full justify-between sm:h-1/2 sm:w-1/4">
        <div class="w-full">
            <h2 class="text-sm font-medium text-gray-600 mb-1">Correos spam detectados</h2>
            <p class="text-lg font-semibold text-gray-600 mb-1">{{cantSpam}}</p>
        </div>
        <div class="w-1/4 flex justify-center items-center">
            <div class="bg-orange-600 rounded-full w-12 h-12 flex items-center justify-center">
                <i class="fa-solid fa-hat-cowboy text-white text-2xl"></i>
            </div>
        </div>
    </div>
</div>
```

Fig. 58. Plantilla HTML para presentar el componente de las cartas con los datos detectados.

Finalmente, la siguiente imagen ilustra la interfaz que verá el usuario, proporcionándole estadísticas de spam, correos normales, alertas enviadas al usuario y correos reportados.

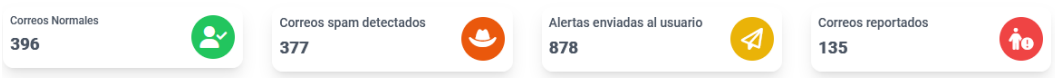


Fig. 59. Componente de las cartas que presentan datos de detección.

Explicado el procedimiento para administrar las vistas y los datos mostrados, en la siguiente gráfica se ilustra la interfaz completa del dashboard, mostrando gráficas

estadísticas del monitoreo del agente, en dónde se podrá visualizar a cantidad de correos detectados como normales, spam, alertas enviadas y alertas reportadas durante el período de los últimos siete días.

Fig. 60. Interfaz web del dashboard.

Dentro del menú Historial, se podrá observar todos los correos electrónicos que han sido reportados por el usuario, donde el administrador puede seleccionar uno y ver los detalles de este.

Fig. 61. Información detallada sobre el correo reportado por el usuario, dentro de un entorno controlado.

En el menú de filtros, el administrador puede aplicar dos tipos de listas, tanto blancas como negras, en donde la primera permitirá que la dirección de correo electrónico que envíe correos no sea analizada por el agente

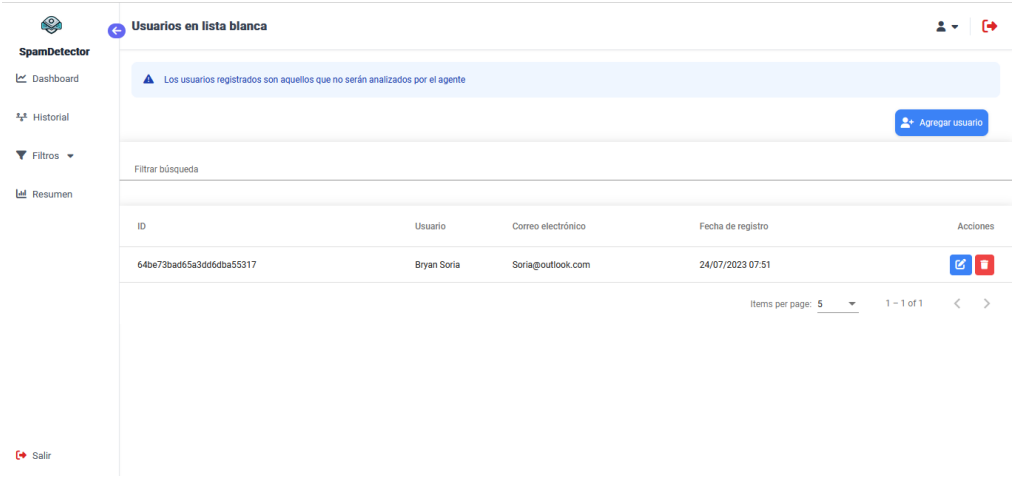


Fig. 62. Interfaz de usuarios registrados en lista blanca en un entorno controlado.

Las listas negras permitirán al administrador añadir cuentas identificadas como emisoras de spam, en donde cuando se registren en el filtro automáticamente todos los correos que la dirección de correo electrónico envíe serán removidas a la carpeta de spam del usuario receptor.

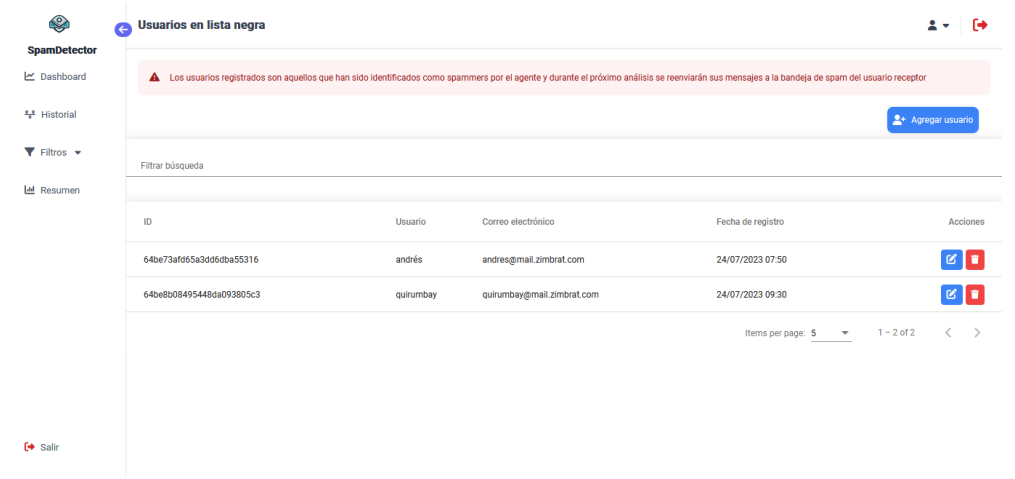


Fig. 63. Interfaz de usuarios registrados en lista negra en un entorno controlado.

Esto permitirá al agente evitar falsos negativos ya que el usuario que envíe spam, pero el agente no logre detectarlo como tal, este puede pasar por desapercibido y llegar a la bandeja de entrada del usuario.

De esta manera se aplica otro filtro a la detección de spam y hacer que el agente pueda ser más efectivo al momento de detectar un correo electrónico no deseado.

La implementación de tailwindcss ayudó a realizar un diseño responsivo de manera más rápida comparado con usar css puro, enfocandose en el pleno desarrollo del proyecto, pero sin dejar de lado la parte visual del usuario

Fig. 64. Interfaz web responsiva del dashboard.

2.5.2.3.5. Despliegue

Para las pruebas del correcto funcionamiento del ambiente web, se lleva a cabo la ejecución de un servidor web local ejecutado por NodeJS, permitiendo alojar la aplicación y simular un entorno de producción real, probando las funcionalidades de manera segura y controlada.

2.5.3. Metodología CSF

La metodología brinda una serie de pasos recomendados para obtener un mejor resultado al momento de detectar los riesgos dentro de la organización que se esté aplicando.

2.5.3.1.Alcance

Para poder determinar el alcance se realizó una revisión de las metas y objetivos de las unidades administrativas del GAD municipal, puntualmente del departamento de TI, donde se resaltan a continuación.

Tabla 24 - Metas y objetivos del GAD municipal dentro del departamento de TI.

Unidad	Objetivo	Indicador
Sistemas y Recursos Tecnológicos	Coordinar y administrar eficientemente los recursos tecnológicos informáticos, mediante la utilización de tecnologías de información y la automatización de procesos, a través de la web institucional, servicios informáticos, redes, equipos de computación para el procesamiento automático de datos, acceso a la información y seguridad de los sistemas informáticos, a fin de apoyar de manera eficaz el desarrollo tecnológico y gestión municipal y la toma de decisiones en beneficio de la colectividad del cantón.	• Gestión Administrativa de la Unidad. • Gestión y mantenimiento de Hardware • Administración y mantenimiento de Software. • Gestión Interna de Seguridad Informática • Gestión Interna de Redes y comunicaciones. • Gestión Desarrollo de Sistemas y Aplicaciones Informáticas.

El alcance planteado se basa en la meta planteada “seguridad de los sistemas informáticos” del indicador “Gestión interna de Seguridad Informática”, será el desarrollo de una gente especializado para la supervisión del correo electrónico,

donde tendrá la capacidad de detectar y alertar posibles correos electrónicos que representen contenido spam.

2.5.3.2.Orientación

Se tendrán en cuenta los siguientes parámetros para poder determinar las probabilidades de ocurrencia e impacto de un riesgo dentro del servidor de correo electrónico. En la siguiente tabla, se plantearán ciertos niveles de probabilidad de un riesgo acompañado de su descripción.

Tabla 25 - Matriz de evaluación de probabilidad de ocurrencia.

Probabilidad	**Nivel**	**Descripción**
Muy Baja	1	Extremadamente poco probable que ocurra.
Baja	2	Poco probable que ocurra en circunstancias normales.
Moderada	3	Puede ocurrir en ciertas circunstancias, pero poco frecuente
Alta	4	Probable que ocurra en circunstancias normales o en situaciones específicas
Muy alta	5	Es probable que ocurra en la mayoría de las circunstancias o en la mayoría de las situaciones

Establecido la probabilidad de ocurrencia de un riesgo, se deberá determinar cuál es el impacto de este dentro de la organización, para aquello se seguirá la siguiente matriz de impacto junto con su descripción respectiva.

Tabla 26 - Matriz de evaluación de impacto del riesgo.

Impacto	**Nivel**	**Descripción**
Muy Baja	1	Impacto insignificante y altamente improbable que cause daños significativos
Baja	2	Impacto mínimo y poco probable que cause daños significantes.
Moderada	3	Puede generar impactos negativos, pero se pueden gestionar sin mayor dificultad.

Impacto	Nivel	Descripción
Alta	4	Puede generar impactos significativos y se requieren medidas adecuadas para gestionar los riesgos.
Muy alta	5	Puede generar impactos irreversibles y no hay medidas que reestablezcan el servicio.

La matriz para determinar la probabilidad y el impacto para determinar el análisis cuantitativo de los riesgos identificados será importante para poder establecer los niveles objetivos de cada una de ellas, de tal forma que se pueda analizar las brechas de manera más eficiente.

Tabla 27 - Matriz de probabilidad e impacto.

Probabilidad		Nivel de Impacto				
		Muy Baja	Baja	Medio	Alta	Muy Alta
		1	2	3	4	5
Probabilidad de Ocurrencia	Muy Alta	5	10	15	20	25
	Alta	4	8	12	16	20
	Medio	3	6	9	12	15
	Baja	2	4	6	8	10
	Muy Baja	1	2	3	4	5

Tabla 28 - Niveles de riesgos.

Nivel de Riesgo	Valor	Concepto
Muy Bajo	1	Es un riesgo extremadamente bajo y se toman las medidas según sea necesario,
Bajo	2	Es un riesgo con una probabilidad e impacto de nivel bajo, donde las medidas a implementar no son urgentes.

Nivel de Riesgo	Valor	Concepto
Medio	3-8	Es un riesgo con una probabilidad e impacto importante, donde se requieren implementar las medidas adecuadas.
Alto	9-12	Es un riesgo con una probabilidad e impacto alto, donde las medidas se deben de plantear urgentemente
Muy Alto	15-25	Es un riesgo con una probabilidad e impacto muy alto, donde las medidas a implantar deben de ser urgentes y contundentes.

Una vez definida los diferentes parámetros sobre la ocurrencia e impacto de un riesgo, se procederá a calcular de manera cuantitativa el nivel de riesgo con la siguiente formula:

Nivel de riesgo = NP*NI

Donde:

NP: Nivel de probabilidad

NI: Nivel de impacto

Una vez definido los parámetros de evaluación del riesgo, se procederá a establecer las actividades a realizar para identificar los posibles riesgos junto con los diferentes parámetros establecidos previamente, para ello se seguirá los pasos que establece el núcleo de la metodología CSF de NIST, donde detallan las funciones, categorías y subcategorías a analizar.

Se procederá a seguir las funciones que establece el marco de trabajo CSF, donde se seleccionaron las siguientes categorías y subcategorías que permitirán realizar una evaluación e identificar los niveles de cada uno, donde la selección de cada categoría y subcategoría se basó en el alcance del proyecto, es decir, en el servidor de correo electrónico (Anexo 17). Durante esta sección, se establecerán preguntas a realizar al administrador de los servidores del GAD municipal sobre la gestión de

riesgos general y específicamente del servidor del correo electrónico, para poder determinar los riesgos que son más susceptibles a ocurrir.

Tabla 29 – Categorías y subcategorías del componente núcleo a analizar.

Identificador de función	Función	Identificador único de subcategoría	Categoría
ID	Identificar	ID.AM	Gestión de activos
		ID.RA	Evaluación de riesgos
		ID.RM	Estrategia de gestión de riesgos
PR	Proteger	PR.AC	Gestión de identidad y control de acceso
		PR.DS	Seguridad de datos
DE	Detectar	DE.AE	Anomalías y eventos
RS	Responder	RS.AN	Análisis
		RS.MI	Mitigación
RC	Recuperar	RC.RP	Planificación de recuperación

Seleccionado las categorías a usar en esta fase, se procederán a definir las actividades para poder determinar el estado y nivel de cada una de ellas por cada función.

Identificar

Tabla 30 - Actividades a realizar para obtener información de la categoría identificar.

Categoría	ID subcategoría	Actividad
Gestión de activos	ID.AM-2	1. Revisar la gestión de riesgos actual del departamento. 2. Revisar el plan organizacional actual
Evaluación de riesgos	ID.RA-1	1. Identificar los activos dentro del departamento 2. Identificar si existe un análisis de riesgo de los activos

Categoría	ID subcategoría	Actividad
Estrategia de gestión de riesgos	ID.RM-1	Identificar si existe un responsable de la gestión de riesgos del servidor de correo electrónico

Proteger

Tabla 31 - Actividades a realizar para obtener información de la categoría proteger.

Categoría	ID subcategoría	Actividad
Gestión de identidad, autenticación y control de acceso	PR.AC-1	Identificar si existen políticas de seguridad respecto a las identidades y credenciales para los usuarios, y como se emiten y administran
	PR.AC-2	Identificar si existen normas de acceso al servidor de correo electrónico de manera física.
	PR.AC-4	Identificar si el acceso al servidor de correo electrónico establece acceso mínimo, tanto física como remotamente.
Seguridad de los datos	PR.DS-2	Identificar si la comunicación entre el servidor y los clientes se encuentra protegida por protocolos.
	PR.DS-4	Identificar si el servidor aplica métodos de disponibilidad
	PR.DS-5	Identificar si se aplican métodos de filtrado de datos

Detectar

Tabla 32 - Actividades a realizar para obtener información de la categoría detectar.

Categoría	ID categoría	Actividad
Anomalías y Eventos	DE.AE-5	Identificar si se aplican herramientas de alertas en caso de incidentes

Responder

Tabla 33 - Actividades a realizar para obtener información de la categoría responder.

Categoría	ID categoría	Actividad
Análisis	RS.AN-1	1. Identificar si se analizan las notificaciones de los sistemas de detección
	RS.AN-2	1. Identificar si se comprende el impacto de tener spam en los correos electrónicos
Mitigación	RS.MI-1	1. Identificar si se toma medidas para evitar la propagación del incidente
	RS.MI-2	1. Identificar si se toman acciones inmediatas para mitigar los efectos del incidente

Recuperar

Tabla 34 - Actividades a realizar para obtener información de la categoría recuperar.

Categoría	ID categoría	Actividad
Planificación de la recuperación	RC.RP-1	1. Identificar si existe un plan de recuperación en el caso de un incidente en el servidor.

Para poder cumplir con las actividades, se plantearon entrevistas por cada función al administrador del servidor de correo electrónico, donde cada pregunta fue específicamente enfocada a este servicio.

2.5.3.3.Crear un perfil actual

Para poder establecer el perfil donde se encuentra actualmente el departamento de TI, se realizaron varias entrevistas donde se tuvieron en consideración las gestiones de riesgos actuales que poseen para poder identificar, proteger, detectar, responder y recuperar los servicios frente a un riesgo, donde se pudo establecer el siguiente nivel de manera general.

Luego de haber realizado las entrevistas y el método de observación, se pudo plasmar un diagrama de red del servidor de correo electrónico, dónde admite correos electrónicos a dominios externos, además se puede entrar al correo fuera de la institución. Por otro lado, solo se cuenta protegido por un firewall que dentro de él se aplican reglas básicas, además de contar con el servidor virtualizado.

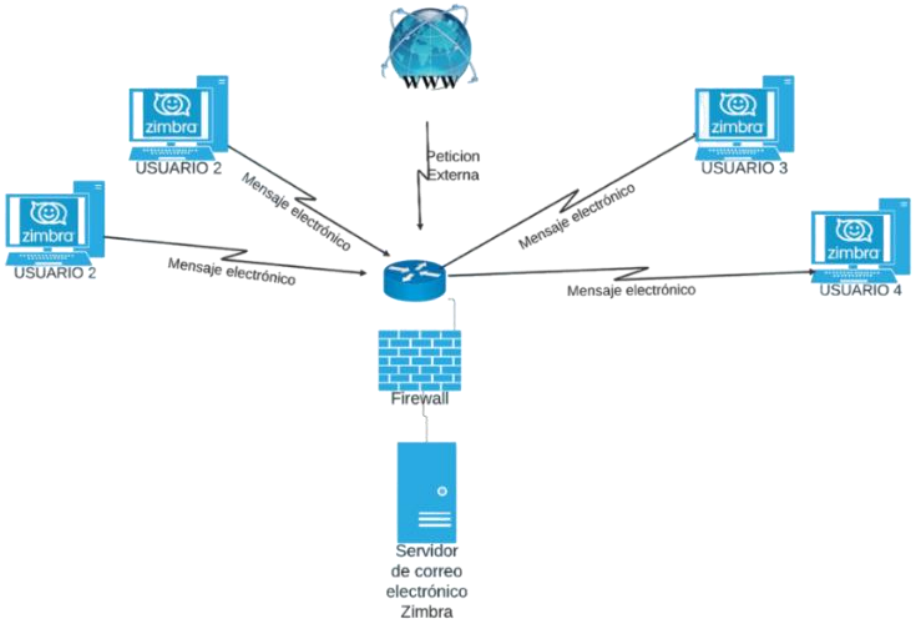


Fig. 65. Topología de red del servidor de correo electrónico.

Determinado cada nivel (Anexo 18) de las categorías y subcategorías de las diferentes funciones, se pudieron observar en el Anexo 19 que la mayoría de los

niveles son de parcial y riesgo informado, es decir no existe ningún tipo de políticas formales sobre los riesgos informáticos en el servidor de correo electrónico.

Analizado los niveles, existen nueve en nivel parcial y seis en estado riesgo informado, por lo que se estableció un nivel general por todas las funciones.

Tabla 35 - Nivel actual promedio del GAD municipal en cuanto al servidor de correo electrónico.

N°	Nivel	Descripción
1	Parcial	Dentro del departamento del TI, puntualmente en el servicio de correo electrónico, no se aplican gestiones de riesgos formales, pero se aplican políticas a un nivel parcial, es decir, son las que ya vienen por defecto en el software Zimbra y del mismo sistema operativo. Durante un incidente, todo se lo realiza en tiempo real, es decir no aplican un mecanismo para poder detectar un incidente dentro del servidor de correo electrónico

2.5.3.4.Análisis de riesgos

Mediante el análisis de riesgo plantea como objetivo determinar las probabilidades de que un riesgo suceda, permitiendo la evaluación las consecuencias que esta pueda ocasionar, involucrando la recolección y análisis de información para determinar el impacto del riesgo [115].

Para poder identificar los riesgos dentro del servidor de correo electrónico, se debe de establecer los activos que éste posee, dónde se tendrán en cuenta los siguientes:

- Datos
- Hardware
- Software
- Comunicaciones
- Personal

Tabla 36 -Tipos de activos [116].

Tipos de activos	Descripción
Servicio	Conjunto de servicios necesarios para la gestión y funcionamiento de los sistemas de información.
Software	Aplicaciones y programas informáticos que permiten administrar y procesar datos
Hardware	Equipos informáticos que permiten alojar datos, servicios y aplicaciones.
Comunicaciones	Redes de comunicación que permiten intercambiar los datos
Soportes de Información	Medios físicos usados para almacenar y respaldar datos.
Instalaciones	Infraestructura donde se alojan los sistemas de información.
Personal	Personas encargadas de la administración, gestión y control de los sistemas de información

La recolección de información mediante la información pública de GAD municipal, método de observación y entrevistas realizada al administrador de TI, se pudo obtener los siguientes activos según la categoría definida anteriormente.

Tabla 37 - Activos identificados del servicio de correo electrónico.

Categoría	Identificador	Activo	Descripción
Servicio	serv01	Zimbra web	Página web que permite enviar y recibir mensajes de correo electrónico dentro de la organización
Software	soft01	Zimbra Collaboration Suite	Es una suite de Zimbra que incluye servidor de correo electrónico, calendario, contactos, mensajería, etc.

Categoría	Identificador	Activo	Descripción
Hardware	hard01	Servidor virtualizado	Servidor físico que se ha virtualizado para alojar el servidor de correo electrónico
Comunicaciones	com01	Zimbra web	Por medio de este servicio se envían y reciben correos electrónicos
Soporte de información	sopt01	Almacenamiento de correos	Medio de almacenamiento de los correos electrónicos y archivos adjuntos
Instalaciones	inst01	Centro de datos	Espacio físico que alberga el servidor de correo electrónico
Personal	pers01	Administrador del servidor	Responsable de gestionar y mantener el servicio de correo electrónico

Identificado los activos del servidor de correo electrónico, realizando las actividades para poder identificar los niveles de cada subcategoría y establecer los niveles de estos, en el Anexo 20 se pueden identificar los riesgos por cada función mediante las justificaciones de cada nivel.

Además, en los Anexos 3, 4, 5 y 6 se pudo evidenciar mediante el método de observación la existencia de varios correos electrónicos categorizados como spam dentro del GAD.

2.5.3.5.Creación de un perfil objetivo

Para poder establecer este nivel, se tuvieron en cuenta varios factores, cómo el nivel actual y el nivel objetivo planteado en esta sección, donde pasar del nivel uno al tres, implica una brecha de dos niveles, en la que se debe de tener en cuenta varios factores como el personal, economía y recursos computacionales.

Además, la justificación de no optar por el nivel cuatro, es que la misma metodología no recomienda dar el salto desde el nivel uno hacia el rango más alto, ya que implica los factores mencionados anteriormente sumando con el tiempo de ejecución. Otro factor importante para considerar es que el nivel cuatro sea adaptativos a cada servicio, realizando análisis mediante indicadores predictivos.

Tabla 38 - Nivel general de las categorías.

N°	Nivel	Descripción
3	Repetible	El objetivo de este nivel es que la mayoría de las subcategorías que se requieran subir a este nivel, implementen formalmente las políticas de ciberseguridad de manera formal y que éste sea actualizado cada vez que lo requiera

2.5.3.6.Plan de acción

En este paso propuesto por la metodología, se recomienda realizar el plan de acción detallado en el Anexo 21, donde cada recomendación está alineada a las cinco funciones del componente núcleo, subcategorías, perfil actual, perfil objetivo y los riesgos encontrados en la etapa de Análisis de riesgo.

Una vez implementado el agente detector de spam en el servicio de Zimbra del GAD Anexo 23, se puede evidenciar que el agente está monitoreando los correos electrónicos entrantes en el servicio de Zimbra, específicamente configurado para los dominios externos del GAD, donde durante esa ejecución estaba detectando solo correos con el dominio propio.

Durante el Anexo 24, se puede apreciar que se han detectados varios correos electrónicos externos que contienen mensajes normales y spam, pero por temas de privacidad no se puede saber a quién le está llegando este tipo de correos, es por eso, que en el Anexo 25, se hizo una prueba enviando un spam hacia el

administrador de TI mediante su autorización por medio de un correo electrónico con dominio Outlook, donde se puede apreciar que el agente lo detectó como spam y lo removió a la carpeta correspondiente Anexo 26, posteriormente el agente le envió una alerta al usuario para que este pueda reportarlo, detallando el remitente del correo.

En el Anexo 27, se puede apreciar la interfaz para que el usuario alertado pueda reportar el correo electrónico, pero antes de esto deberá leer los términos para que sepa lo que se hará y no con el correo electrónico reportado.

En el Anexo 28 se puede apreciar que el agente removió un correo electrónico categorizado como spam desde un dominio externo, logrando mitigar que el usuario pueda abrir este tipo de correos electrónicos.

En el Anexo 30 se evidencia el correo electrónico que ha sido reportado por el usuario, mostrando detalles como el Id del mensaje, remitente, receptor, asunto, mensaje y otros aspectos importantes como la fecha.

2.5.4. Arquitectura del sistema

La siguiente gráfica representa la arquitectura del sistema del agente detector de spam, en la cual estará instalado en el servidor de correo electrónico moniroreando constantemente en busca de correos categorizados como spam

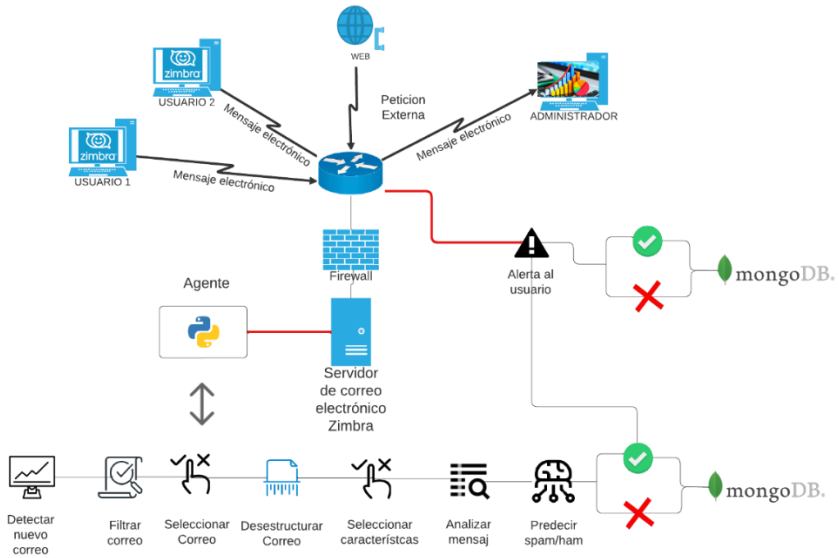


Fig. 66. Arquitectura del sistema.

2.5. Resultados

El desarrollo del agente para la detección de spam en correos electrónicos en el servicio de Zimbra tiene como resultados los siguientes puntos:

- El agente puede ser implementado en diferentes sistemas operativos que sean distribuciones de Linux que empleen el servicio de Zimbra y que cuente con las versiones de las librerías correctamente.
- El agente puede ser ejecutado como un subproceso del sistema operativo, por el cual puede estar analizando los correos electrónicos sin tener que estar ejecutando manualmente el script.
- El agente puede adoptar cualquier algoritmo de aprendizaje automático para la detección de spam.
- El agente puede remover el correo electrónico detectado como spam desde la bandeja de entrada hacia la carpeta de spam para cada destinatario que contenga el mensaje de correo.
- El agente puede enviar alertas a todos los usuarios que sean receptores en un correo electrónico
- El agente puede detectar dominios externos al propio servidor de Zimbra.
- El administrador del servidor puede a través de la interfaz web desarrollada ver gráficas relacionadas a la detección de correos normales, spam, alertas enviadas a usuarios y las alertas que han sido reportadas.
- El administrador puede a través de la interfaz gráfica añadir a correos electrónicos en la lista blanca o la lista negra.
- El administrador puede generar un documento PDF como XLSX para la toma de decisiones frente al servicio de correo electrónico.

2.6.1. Resultados de la variable

Variable: Cantidad de correos electrónicos detectados como spam antes y después de su implementación en el servicio de Zimbra.

2.6.1.1. Entorno controlado

2.6.1.1.1. Precisión del sistema antispam de Zimbra

Para poder establecer la precisión, se enviaron alrededor de 300 correos electrónicos, entre ellos 150 correos normales y 150 no deseados, donde el detector por defecto que incluye Zimbra en su sistema no logró detectar ningún correo spam.

Tabla 39 - Detección de correos normales y spam por el sistema antispam de Zimbra.

Tipo	Enviados	Analizados	Falsos positivos
ham	150	0	0
spam	150	0	0

En este proceso, se procedió a remover un correo electrónico manualmente marcándolo como spam Anexo 31. En el Anexo 32 se puede observar la puntuación del correo marcado como no deseado, donde fue de 0.171 y para que sea considerado como spam, la calificación debe de ser de al menos 6.6, por lo que la diferencia es muy amplia y no precisa.

2.6.1.1.2. Precisión del agente antispam desarrollado en un entorno controlado

En esta etapa, se procedió a enviar los mismos correos electrónicos de la etapa anterior, donde se obtuvieron los siguientes resultados.

Tabla 40 - Detección de correos normales y spam después de implementar el agente en el servicio de Zimbra.

Tipo	Enviados	Analizados	Positivos	Falsos positivos	Precision
ham	150	150	146	4	97%
spam	150	150	147	3	98%

La implementación del agente permitió analizar el 100% de correos electrónicos clasificados como spam, a diferencia del 0% por el sistema antispam de Zimbra.

El agente demostró una precisión del 98% para la detección de spam, mientras que para detectar correos normales fue de 97% aproximadamente, demostrando una mejora con la implementación del agente detector de spam en el servicio controlado.

2.6.1.1.3. Simulación de entorno de producción

Durante la fase de desarrollo se procedió a enviar contantemente correos normales y no deseados durante un lapso se cinco días, donde se pudo obtener las siguientes características de manera general que podrá ser analizada junto a la versión de producción.

Tabla 41 - Envío de ham y spam en un entorno controlado.

Día	**Correos analizados**	**ham**	**spam**	**Alertas enviadas**	**Correos reportados**
18/07/2023	70	65	5	7	4
19/07/2023	66	60	5	13	10
20/07/2023	74	63	7	15	5
21/07/2023	66	58	6	12	6
22/07/2023	79	69	10	18	7
Total	355	315	33	65	32

2.6.2.1. Entorno de producción

En esta etapa, no se puede identificar si se han detectados positivos o faltos positivos ya que es un entorno donde los datos son confidenciales, a diferencia del entorno de desarrollo que se podía analizar cada correo enviado.

Dicho esto, se realizaron pruebas con el agente en ejecución durante varios días, obteniendo los siguientes resultados.

Tabla 42 - Datos recolectados por el agente durante la implementación en el entorno de producción.

Día	Correos analizados	ham	spam	Alertas enviadas	Correos reportados
26/07/2023	45	34	11	18	0
27/07/2023	153	136	17	17	1
31/07/2023	218	202	16	18	2
1/08/2023	444	417	27	27	0
2/08/2023	303	292	11	11	0
Total	1163	1081	82	91	3

CONCLUSIONES

- La metodología KDD ayudó a obtener un conjunto de datos que incluye correos spam y ham, lo que permitió entrenar de manera eficiente cada algoritmo seleccionado.
- Mediante una comparación de resultados se determinó que el algoritmo de aprendizaje automático con mejores resultados fue Random Forest frente a algoritmos como Decision Tree y Naïve Bayes.
- La implementación del agente en el servidor de correo electrónico Zimbra mediante un entorno controlado permitió detectar todos los correos que se transmitían en tiempo real y los categorizados como spam removerlos a la carpeta correspondiente.
- La implementación del agente en el servidor de correo electrónico Zimbra del GAD permitió detectar que los correos categorizados como spam provenían de dominios externos
- El medio alertivo permitió detectar la cantidad de correos que se transmiten en el servidor de correo electrónico, clasificándolos como ham, spam, alertas enviadas a los usuarios y la cantidad de correos reportados.
- El dashboard web depende de que un usuario reporte un correo como spam para que el administrador pueda analizar dicho mensaje.

RECOMENDACIONES

- Se recomienda que se instalen las librerías con las versiones específicas para el correcto funcionamiento del agente detector de spam.
- Se recomienda que el agente se ejecute en segundo plano para el monitoreo constante de los correos electrónicos.
- Se sugiere que se en un entorno de producción se comunique a los usuarios sobre las alertas generadas por el agente y que, si se trata de un spam, que este sea reportado para poder ser analizado posteriormente.
- Se recomienda el administrador añada a la lista negra aquellas direcciones de correo electrónico que han sido reportados como spam.
- Se recomienda añadir direcciones de correo electrónico a la lista blanca si desea en el proceso de análisis.
- Se recomienda que el administrador configure los dominios permitidos dentro del servicio de Zimbra mediante reglas del mismo servicio
- Se recomienda que para el correcto funcionamiento de la interfaz que permite al usuario reportar el correo electrónico como spam cuenta con un certificado SSL.

REFERENCIAS

[1] H.-K. Lee, "A Study on Hacking E-Mail Detection using Indicators of Compromise," pp. 1–8, Sep. 2020, doi: 10.33778/kcsa.2020.20.3.021.

[2] Kapersky, "Informe de Kaspersky sobre el spam y el phishing en el año 2021 | Securelist," Feb. 09, 2022. https://securelist.lat/spam-and-phishing-in-2021/96171/ (accessed Nov. 15, 2022).

[3] "La Libertad," Sep. 2022. http://www.lalibertad.gob.ec/?menu=31 (accessed Nov. 16, 2022).

[4] G. Chiquito, "UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES CARRERA DE INFORMÁTICA TRABAJO DE TITULACIÓN," UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA, La Libertad, 2016.

[5] Carlos Nicolas, "EMAIL SPAM DETECTION USING MACHINE LEARNING BASED TEXT ANALYSIS," Universitat Politècnica de Catalunya, Barcelona, 2022. Accessed: Aug. 06, 2023. [Online]. Available:

https://upcommons.upc.edu/bitstream/handle/2117/386385/EMAIL%20SPAM%20DETECTION%20USING%20MACHINE%20LEARNING%20BASED%20TEXT%20ANALYSIS.pdf?sequence=6

[6] M. León and N. Sotelo, “Detección de Correos Falsos y Detección de Intrusos,” Detección de Correos Falsos y Detección de Intrusos, Bogotá, 2020.

[7] D. ALBÁN, “ANÁLISIS Y DISEÑO DE UN MODELO PREDICTIVO PARADETECCIÓN DE PHISHING BASADO EN URL Y CORPUS DELCORREO ELECTRÓNICO,” Apr. 2022. https://bibdigital.epn.edu.ec/bitstream/15000/22525/1/CD 12024.pdf (accessed Dec. 20, 2022).

[8] S. Timarán, I. Hernández, S. Caicedo, A. Hidalgo, and J. Alvarado, “El proceso de descubrimiento de conocimiento en bases de datos,” *Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional*, pp. 63–86, 2016, doi: 10.16925/9789587600490.

[9] OMSTD Project, “Bienvenido a OMSTD (Open Methodology for Security Tool Developers) — OMSTD - Open Methodology for Security Tool Developers :: Documentation.” https://omstd.readthedocs.io/ (accessed May 20, 2023).

[10] A. Mahn, J. Marron, S. Quinn, and D. Topper, “Getting Started with the NIST Cybersecurity Framework: A Quick Start Guide What is the NIST Cybersecurity Framework, and how can my organization use it? NIST Special Publication 1271,” Aug. 2021, doi: 10.6028/NIST.SP.1271.

[11] “Documentation for Visual Studio Code.” https://code.visualstudio.com/docs (accessed Nov. 23, 2022).

[12] “What is Python? Executive Summary | Python.org.” https://www.python.org/doc/essays/blurb/ (accessed Nov. 23, 2022).

[13] “pandas - Python Data Analysis Library.” https://pandas.pydata.org/ (accessed May 04, 2023).

[14] “Welcome to Flask — Flask Documentation (2.3.x).” https://flask.palletsprojects.com/en/2.3.x/ (accessed Apr. 26, 2023).

[15] MDN mozilla, “JavaScript | MDN.” https://developer.mozilla.org/es/docs/Web/JavaScript (accessed Jun. 20, 2023).

[16] “Angular - What is Angular?” https://angular.io/guide/what-is-angular (accessed May 04, 2023).

[17] “Angular Material UI component library.” https://material.angular.io/ (accessed May 04, 2023).

[18] “TypeScript: JavaScript With Syntax For Types.” https://www.typescriptlang.org/ (accessed May 04, 2023).

[19] “Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.” https://tailwindcss.com/ (accessed Apr. 26, 2023).

[20] Chartjs, “Chart.js | Chart.js.” https://www.chartjs.org/docs/3.9.1/ (accessed Jul. 27, 2023).

[21] “NgRx - @NGRX/STORE.” https://ngrx.io/guide/store (accessed Jul. 31, 2023).

[22] Universidad de Alcalá, “Scikit-Learn, herramienta básica para el Data Science en Python.” https://www.master-data-scientist.com/scikit-learn-data-science/ (accessed Dec. 20, 2022).

[23] “smtplib.” https://docs.python.org/3/library/smtplib.html (accessed Feb. 09, 2023).

[24] “Zimbra Support.” https://www.zimbra-support.net/index.php/es/ (accessed Dec. 20, 2022).

[25] “¿Qué es VMware Workstation? | Preguntas frecuentes | LATAM.” https://www.vmware.com/latam/products/workstation-pro/faq.html (accessed Dec. 19, 2022).

[26] “The CentOS Project.” https://www.centos.org/ (accessed Dec. 19, 2022).

[27] “¿Qué Es MongoDB? | MongoDB.” https://www.mongodb.com/es/what-is-mongodb (accessed Feb. 09, 2023).

[28] readthedocs, “PyMongo 4.4.1 documentation.” https://pymongo.readthedocs.io/en/stable/ (accessed Jul. 31, 2023).

[29] “xlsx - npm.” https://www.npmjs.com/package/xlsx (accessed Jul. 31, 2023).

[30] “jspdf - npm.” https://www.npmjs.com/package/jspdf (accessed Jul. 31, 2023).

[31] The Radicate Group, “A TECHNOLOGY MARKET RESEARCH FIRM Email Statistics Report, 2021-2025,” Feb. 2021.

[32] T. Shcherbakova and T. Kulikova, “El spam y el phishing en 2021,” Feb. 08, 2022. https://securelist.lat/spam-and-phishing-in-2021/96171/#:~:text=2%2C62%25).-,Estadísticas%3A Phishing,phishing al menos una vez. (accessed Dec. 11, 2022).

[33] Secretaría Nacional de Planificación, “Plan de creación de oportunidades 2021-2025,” 2021. https://www.planificacion.gob.ec/wp-content/uploads/2021/09/Plan-de-Creación-de-Oportunidades-2021-2025-Aprobado.pdf (accessed Dec. 13, 2022).

[34] R. Hernández, C. Fernández, and M. Baptista, *METODOLOGÍA de la investigación*, 5th ed. 2010. [Online]. Available: https://www.icmujeres.gob.mx/wp-content/uploads/2020/05/Sampieri.Met.Inv.pdf

[35] OMSTD Project, “Bienvenido a OMSTD (Open Methodology for Security Tool Developers) — OMSTD - Open Methodology for Security Tool Developers :: Documentation.” https://omstd.readthedocs.io/ (accessed May 21, 2023).

[36] OMSTD Project, “Conceptos de desarrollo — OMSTD - Open Methodology for Security Tool Developers :: Documentation.” https://omstd.readthedocs.io/develop/index.html (accessed May 21, 2023).

[37] T. Chávez, “Programación Modular,” Oct. 2017.

[38] National Institute of Standards and Technology, “Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1,” 2018, doi: 10.6028/NIST.CSWP.04162018.

[39] “Historia La Libertad,” 2015. http://www.lalibertad.gob.ec/index.php/la-libertad/la-historia

[40] ASAMBLEA NACIONAL, *LEY ORGÁNICA DE PROTECCIÓN DE DATOS PERSONALES*. QUITO: Órgano de la República del Ecuador, 2021. Accessed: May 19, 2023. [Online]. Available: https://www.telecomunicaciones.gob.ec/wp-content/uploads/2021/06/Ley-Organica-de-Datos-Personales.pdf

[41] Registro Oficial Suplemento, *CÓDIGO ORGÁNICO INTEGRAL PENAL, COIP*. 2021. Accessed: Jul. 30, 2023. [Online]. Available: https://www.defensa.gob.ec/

[42] cloudfare, “¿Qué es el correo electrónico? | Definición de correo electrónico | Cloudflare,” 2021. https://www.cloudflare.com/es-es/learning/email-security/what-is-email/ (accessed May 16, 2023).

[43] “¿Qué es el Software Libre? - Proyecto GNU - Free Software Foundation.” https://www.gnu.org/philosophy/free-sw.html (accessed May 24, 2023).

[44] R. M. Stallman, *Software libre para una sociedad libre*, Traficantes de Sueñ…. 2004.

[45] K. ASIF, A. SAMI, S. BHARANIDHARAN, and K. RISHNAN, “Efficient Clustering of Emails Into Spam and Ham: The Foundational Study of a

Comprehensive Unsupervised Framework," *Institute of Electrical and Electronics Engineers Access*, vol. 8, 2020, [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9169622

[46] D. Marca and P. Villarroel, "PROTOTIPO DE FILTRADO ANTI SPAM A TRAVÉS DE UNA NUBE PRIVADA," Universidad Politécnica Salesiana Sede Cuenca, Cuenca, 2020.

[47] Avast, "¿Qué es el spam? Tipos de spam y cómo protegerse | Avast." https://www.avast.com/es-es/c-spam (accessed May 14, 2023).

[48] J. Fruhlinger, "What is phishing? Examples, types, and techniques | CSO Online," Apr. 12, 2022. https://www.csoonline.com/article/2117843/what-is-phishing-examples-types-and-techniques.html (accessed May 14, 2023).

[49] A. Harishakar, R. Kaviya, A. Nigilpriya, and V. Narmatha, "MACHINE LEARNING BASED VOICE E-MAIL SYSTEM VISUALLY IMPAIREDUSING IMAP PROTOCOL," *www.irjmets.com @International Research Journal of Modernization in Engineering*, vol. 3371, doi: 10.56726/IRJMETS34950.

[50] S. Ali, "Performance Evaluation of IMAP and POP3 Protocols Using Optimized Network Engineering Tool (OPNET) MASTER OF ENGINEERING in the Department of Electrical and Computer Engineering," University of Victoria, Pakistan, 2020.

[51] K. Dilip, "Review on SMTP Protocol for E-Mail Systems," *International Journal of Research and Analytical Reviews (IJRAR) www.ijrar.org*, vol. 127, 2018, Accessed: May 29, 2023. [Online]. Available: www.ijrar.org

[52] Oracle, "What Is a Database | Oracle." https://www.oracle.com/database/what-is-database/ (accessed May 14, 2023).

[53] "Bases de datos no relacionales | Bases de datos de gráficos | AWS." https://aws.amazon.com/es/nosql/ (accessed Feb. 21, 2023).

[54] Y. Ocaña-Fernández, L. A. Valenzuela-Fernández, and L. L. Garro-Aburto, "Inteligencia artificial y sus implicaciones en la educación superior," *Propósitos y Representaciones*, vol. 7, no. 2, pp. 536–568, Jan. 2019, doi: 10.20511/pyr2019.v7n2.274.

[55] D. Quirumbay, C. Castillo, and I. Coronel, "Una revisión del aprendizaje profundo aplicado a laciberseguridad," *Revista Científica y Tecnológica UPSE*, vol. 9, pp. 57–65, Jun. 2022.

[56] G. Albornoz, "APLICACIÓN DEL APRENDIZAJE AUTOMÁTICO SUPERVISADO EN EL MANTENIMIENTO PREDICTIVO DE LOS MOTORES ELÉCTRICOS DE INDUCCIÓN EN LAS EMPRESAS

MINERAS DEL PERÚ," UNIVERSIDAD NACIONAL DEL CENTRO DEL PERÚ, 2021.

[57] Google Cloud, "¿Qué es el aprendizaje automático? | Google Cloud | Google Cloud." https://cloud.google.com/learn/what-is-machine-learning?hl=es-419 (accessed May 16, 2023).

[58] G. Ruiz, "Modelo de análisis de datos utilizando técnicas de aprendizaje supervisado y no supervisado, para identificar patrones en la información generada por los pacientes, sometidos a juegos diseñados como un instrumento de apoyo terapéutico," Universidad Jorge Tadeo Lozano Facultad de Ciencias Naturales e Ingeniería, Bogotá, 2019.

[59] I. Joakin, "Aplicación de tecnologías de aprendizaje automático para predecir negocios y toma decisiones empresariales.," UNIVERSIDAD NACIONAL DE LA PLATA, La Plata, 2021.

[60] J. Cárdenas, G. Olivares, and R. Alfaro, "Clasificación automática de textos usando redes de palabras," *Revista signos. Estudios de Lingüística*, vol. 47, no. 86, pp. 346–364, 2014, doi: 10.4067/S0718-09342014000300001.

[61] H. Taherdoost, "Different Types of Data Analysis; Data Analysis Methods and Techniques in Research Projects," *International Journal of Academic Research in Management (IJARM)*, vol. 9, no. 1, 2020, Accessed: May 29, 2023. [Online]. Available: www.elvedit.com

[62] R. Snehkunj and K. Vachiyatwala, "Data Analysis Using Pandas Library of Python," *Acta Scientific COMPUTER SCIENCES*, vol. 4, no. 3, Mar. 2022.

[63] socialcops, *basic data cleaning*.

[64] A. Messina, "Diseño e implementación de una extensión de Chrome para la detección de sitios web de Phishing utilizando aprendizaje automático," 2021, Accessed: May 30, 2023. [Online]. Available: https://repositorio.uam.es/handle/10486/700048

[65] D. Ward, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2015.

[66] "Analyzing machine learning model performance," Mar. 14, 2022. https://cloud.ibm.com/docs/watson-knowledge-studio?topic=watson-knowledge-studio-evaluate-ml (accessed May 30, 2023).

[67] A. Meiriza, E. Lestari, P. Putra, A. Monaputri, and D. Lestari, "Prediction Graduate Student Use Naive Bayes Classifier," *Advances in Intelligent Systems Research*, vol. 172, pp. 370–375, May 2020, doi: 10.2991/AISR.K.200424.056.

[68] Matlab, “Support Vector Machine (SVM) .” https://la.mathworks.com/discovery/support-vector-machine.html (accessed May 30, 2023).

[69] J. Ali, N. Ahmad, and R. Khan, “Random Forests and Decision Trees,” *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 5, pp. 272–278, Sep. 2012, doi: 10.1023/A:1010933404324.

[70] W. Campos and Y. Trujillo, “Software Effort Estimation Using Modified Fuzzy C Means Clustering and Hybrid ABC-MCS Optimization in Neural Network,” *Revista Cubana de Ciencias Informáticas*, vol. 29, no. 1, pp. 251–263, Jun. 2020, doi: 10.1515/JISYS-2017-0121.

[71] Abirami. N, Lavanya. S, and Madhanghi. A, “A Detailed Study of Client-Server and its Architecture,” no. June, Nov. 2019.

[72] V. Gabriela, “ANÁLISIS DE FRAMEWORKS DE DESARROLLO DE API REST Y SU IMPACTO EN EL RENDIMIENTO DE APLICACIONES WEB CON ARQUITECTURA SPA,” UNIVERSIDAD TÉCNICA DEL NORTE, Ibarra, 2018. Accessed: May 30, 2023. [Online]. Available: http://repositorio.utn.edu.ec/bitstream/123456789/8264/1/PG%20659%20TESIS.pdf

[73] Kaspersky, “What is Cyber Security? | Definition, Types, and User Protection.” https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security (accessed May 28, 2023).

[74] IBM, “What is Cybersecurity?” https://www.ibm.com/topics/cybersecurity (accessed May 28, 2023).

[75] Oracle, “What is Machine Learning?” https://www.oracle.com/artificial-intelligence/machine-learning/what-is-machine-learning/ (accessed May 28, 2023).

[76] G. García, “Modelo de Machine Learning para la Clasificación de pacientes en términos del nivel asistencial requerido en una urgencia pediátrica con Área de Cuidados Mínimos,” UNIVERSIDAD TECNOLOGICA DE BOLÍVAR, Cartagena, 2014. Accessed: May 27, 2023. [Online]. Available: https://repositorio.utb.edu.co/bitstream/handle/20.500.12585/1200/0068210.pdf?sequence=1&isAllowed=y

[77] T. Jiang, J. L. Gradus, and A. J. Rosellini, “Supervised Machine Learning: A Brief Primer,” *Behavior therapy*, vol. 51, no. 5, pp. 675–687, Sep. 2020, doi: 10.1016/J.BETH.2020.05.002.

[78] J. Wang and F. Biljecki, “Unsupervised machine learning in urban studies: A systematic review of applications,” vol. 129, Oct. 2022, doi: 10.1016/j.cities.2022.103925.

[79] M. Pina, “Aplicación de técnicas de aprendizaje por refuerzo a navegación visual,” Escuela Politécnica Superior, 2022. Accessed: May 27, 2023. [Online]. Available: https://rua.ua.es/dspace/bitstream/10045/124262/1/TFM-Monica-Pino-Navarro.pdf

[80] M. Wazid, A. K. Das, V. Chamola, and Y. Park, “Uniting cyber security and machine learning: Advantages, challenges and future research,” *ICT Express*, vol. 8, no. 3, pp. 313–321, Sep. 2022, doi: 10.1016/J.ICTE.2022.04.007.

[81] M. Choubisa and D. Upadhyay, “Machine Learning In Cyber Security,” Aug. 2022.

[82] J. R. Méndez, F. Fdez-Riverola, F. Díaz, and J. M. Corchado, “Sistemas inteligentes para la detección y filtrado de correo spam: una revisión,” *Revista Iberoamericana de Inteligencia Artificial*, vol. 11, pp. 63–81, 2007, Accessed: May 16, 2023. [Online]. Available: http://sing.ei.uvigo.es/http://www.infor.uva.es/~fdiazhttp://bisite.usal.es/

[83] T. Guzella and W. Caminhas, “A review of machine learning approaches to Spam filtering,” *Expert Systems with Applications*, vol. 36, 2009, Accessed: May 27, 2023. [Online]. Available: https://doi.org/10.1016/j.eswa.2009.02.037

[84] J. Edmondson, “Advantages of Databases: Why are databases important to businesses?,” *BusinessTech*, Apr. 09, 2022. https://www.businesstechweekly.com/operational-efficiency/data-management/databases-advantages-benefits/ (accessed May 27, 2023).

[85] K. Prasant, “Why and Which Database in Machine Learning, MySQL or MongoDB | by Prasant Kumar | Geek Culture | Medium,” Jun. 30, 2022. https://medium.com/geekculture/why-and-which-database-in-machine-learning-mysql-or-mongodb-d8e43b29aeb2 (accessed May 13, 2023).

[86] Equipo de Expertos de Ciencia y Tecnología de la Universidad Internacional de Valencia, “Programación SQL: para qué sirve y quién la necesita | VIU España,” *Universidad Internacional de Valencia*, Apr. 03, 2019. https://www.universidadviu.com/es/actualidad/nuestros-expertos/programacion-sql-para-que-sirve-y-quien-la-necesita (accessed May 27, 2023).

[87] A. Díaz, “Introducción a las bases de datos NoSQL: comparativa MongoDB vs Cassandra,” Universidad de Alcalá Escuela Politécnica Superior, 2019.

[88] R. Herranz, “BASES DE DATOS NOSQL: ARQUITECTURA Y EJEMPLOS DE APLICACIÓN,” Universidad Carlos III de Madrid, Leganés, 2014.

[89] P. Mayur, H. Akkamahadevi, T. C, and P. Priyadarshini, “A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing — Sharding in MongoDB and its advantages,” *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 325–330, 2017, doi: 10.1109/I-SMAC.2017.8058365.

[90] M. Raza, N. Dilshani, and M. Ali, “A comprehensive review on email spam classification using machine learning algorithms,” *International Conference on Information Networking (ICOIN)*, pp. 327–332, 2021, doi: 10.1109/ICOIN50884.2021.9334020.

[91] D. Mercado, L. Pedraza, and E. Martínez, “Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo,” vol. 13, no. 2, pp. 88–95, Nov. 2015, doi: 10.15665/rp.v13i2.491.

[92] F. J. Yang, “An implementation of naive bayes classifier,” *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, pp. 301–306, Dec. 2018, doi: 10.1109/CSCI46756.2018.00065.

[93] M. A. Díaz Martínez, M. de los A. Ahumada Cervantes, and J. P. Melo Morín, “Árboles de Decisión como Metodología para Determinar el Rendimiento Académico en Educación Superior,” *Revista Lasallista de Investigación*, vol. 18, no. 2, Accessed: May 28, 2023. [Online]. Available: http://www.scielo.org.co/pdf/rlsi/v18n2/1794-4449-rlsi-18-02-94.pdf

[94] “K-means clustering.” https://www.ibm.com/docs/en/db2/11.5?topic=building-k-means-clustering (accessed May 30, 2023).

[95] A. Junnarkar, S. Adhikari, J. Fagania, P. Chimurkar, and D. Karia, “E-mail spam classification via machine learning and natural language processing,” *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, pp. 693–699, Feb. 2021, doi: 10.1109/ICICV50876.2021.9388530.

[96] D. Radev, “CLAIR collection of fraud email,” *ACL Data and Code Repository*, 2008. http://aclweb.org/aclwiki

[97] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam Filtering with Naive Bayes - Which Naive Bayes?,” in *3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA, 2006.*, Mountain View, Jul. 2004. Accessed: Jun. 12, 2023. [Online]. Available: https://www2.aueb.gr/users/ion/data/enron-spam/readme.txt

[98] M. Wiechmann and M. Noppel, “The Enron-Spam dataset preprocessed in a single, clean csv file.,” Nov. 10, 2021. https://github.com/MWiechmann/enron_spam_data/ (accessed Jun. 12, 2023).

[99] R. Croft, Y. Xie, M. Zahedi, · M Ali Babar, and C. Treude, “An Empirical Study of Developers’ Discussions about Security Challenges of Different Programming Languages,” *arxiv*, pp. 1–50, Nov. 2021, Accessed: Jun. 20, 2023. [Online]. Available: https://arxiv.org/abs/2107.13723

[100] R. Horntvedt and T. Åkesson, “Java, Python and Javascript, a comparison,” *for the degree of Bachelor of Science with a major in Computer Science Spring*, 2019. Accessed: Jun. 20, 2023. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1355073/FULLTEXT01.pdf

[101] S. Khoirom, M. Sonia, B. Laikhuram, J. Laishram, and D. Singh, “Comparative Analysis of Python and Java for Beginners,” *International Research Journal of Engineering and Technology*, vol. 7, no. 8, 2020, Accessed: Jun. 20, 2023. [Online]. Available: www.irjet.net

[102] K. Kuk, P. Milic, P. Spalević, and M. Gocic, “Algorithm design in Python for cybersecurity,” 2019, Accessed: Feb. 20, 2023. [Online]. Available: https://www.researchgate.net/publication/336406416_Algorithm_design_in_Python_for_cybersecurity

[103] M. Khan, F. Zaman, M. Adnan, A. Imroz, M. Rauf, and Z. Phul, “Comparative Case Study: An Evaluation of Performance Computation between SQL and NoSQL Database,” *Journal of Software Engineering*, vol. 1, no. 2, pp. 14–23, Feb. 2023, Accessed: Jun. 20, 2023. [Online]. Available: http://sjhse.smiu.edu.pk/sjhse/index.php/SJHSE/article/view/42/7

[104] A. Malik, A. Burney, and F. Ahmed, “A Comparative Study of Unstructured Data with SQL and NO-SQL Database Management Systems,” *Journal of Computer and Communications*, vol. 08, no. 04, pp. 59–71, Apr. 2020, doi: 10.4236/JCC.2020.84005.

[105] D. Nakhare and A. Hatekar, “A Comparative study of SQL Databases and NoSQL Databases for E-Commerce,” *IJRASET*, 2021, doi: https://doi.org/10.22214/ijraset.2021.39263.

[106] R. Sasikala, “A Comparitive Study on Mongo and Cassandra Database For Data Clustering,” *International Journal of Computer Sciences and Engineering*, vol. 6, no. 11, pp. 147–151, 2018, Accessed: Jun. 20, 2023. [Online]. Available: www.ijcseonline.org

[107] C. Győrödi, R. Gyorodi, and B. Livia, “A Comparative Study Between the Capabilities of MySQl Vs. MongoDB as a Back-End for an Online Platform,” *Article in International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016, doi: 10.14569/IJACSA.2016.071111.

[108] F. Versaci and G. Busonera, “Scaling deep learning data management with Cassandra DB,” *IEEE International Conference on Big Data (Big Data*, pp. 5301–5310, 2022, doi: 10.1109/BigData52589.2021.9672005.

[109] J. N. Castillo, J. R. Garcés, M. P. Navas, and D. F. Segovia, “Base de Datos NoSQL: MongoDB vs. Cassandra en operaciones CRUD (Create, Read, Update, Delete,” *Revista Publicando*, vol. 4, no. 11(1), pp. 79–107, Jun. 2017, Accessed: Jun. 21, 2023. [Online]. Available: https://revistapublicando.org/revista/index.php/crv/article/view/398

[110] D. Ghimire, “Comparative study on Python web frameworks: Flask and Django,” *etropolia University of Applied Sciences*, 2020, [Online]. Available: https://www.theseus.fi/handle/10024/339796

[111] N. Idris, C. Mohd, and P. Shamala, “A Generic Review of Web Technology: DJango and Flask,” *nternational Journal of Advanced Computing Science and Engineering*, vol. 2, no. 1, pp. 34–40, 2020, [Online]. Available: http://download.garuda.kemdikbud.go.id/article.php?article=3380721&val=29650&title=A Generic Review of Web Technology DJango and Flask

[112] M. Lathkar, “Introduction to FastAPI. In: High-Performance Web Apps with FastAPI.,” *Apress, Berkeley, CA*, 2023, [Online]. Available: https://doi.org/10.1007/978-1-4842-9178-8_1

[113] E. Saks, “JavaScript frameworks: Angular vs React vs Vue.” 2019. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/261970/Thesis-Elar-Saks.pdf

[114] J. Salomaa, “Evaluating JavaScript frameworks,” *Computer Science*, 2020, [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1461878/FULLTEXT01.pdf

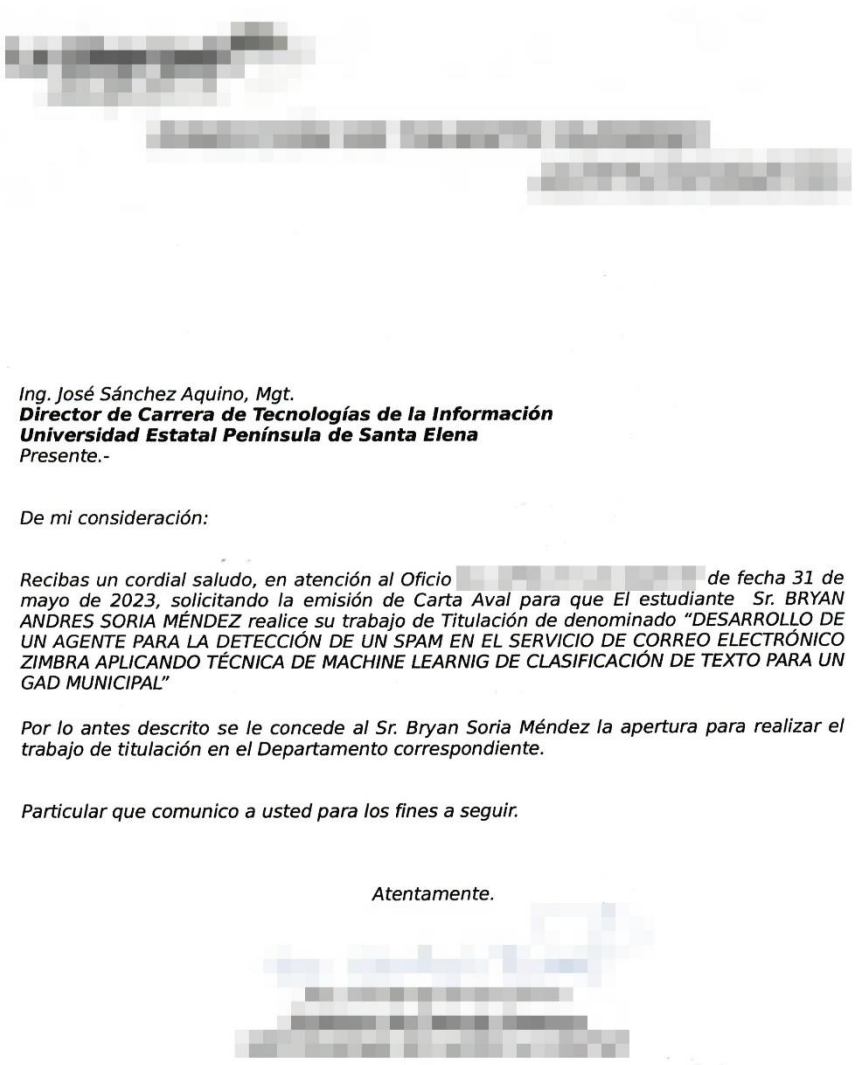
[115] L. Vilcarromero and V. Evit, “Propuesta de implementación de un modelo de gestión de ciberseguridad para el centro de operaciones de seguridad (SOC) de una empresa de telecomunicaciones,” UNIVERSIDAD

PERUANA DE CIENCIAS APLICADAS , Lima. Accessed: Jun. 25, 2023. [Online]. Available: http://hdl.handle.net/10757/624832

[116] A. Paltán, “Evaluación de riesgos y Desarrollo de un plan de recuperación ante desastres informáticos aplicado al Centro de Datos y Comunicaciones de la UPSE,” Universidad Estatal Península de Santa Elena, La Libertad, 2017.

ANEXOS

Anexo 1. Carta de permiso concedido para desarrollar el proyecto dentro del GAD municipal.



Ing. José Sánchez Aquino, Mgt.
Director de Carrera de Tecnologías de la Información
Universidad Estatal Península de Santa Elena
Presente.-

De mi consideración:

Recibas un cordial saludo, en atención al Oficio de fecha 31 de mayo de 2023, solicitando la emisión de Carta Aval para que El estudiante Sr. BRYAN ANDRES SORIA MÉNDEZ realice su trabajo de Titulación de denominado “DESARROLLO DE UN AGENTE PARA LA DETECCIÓN DE UN SPAM EN EL SERVICIO DE CORREO ELECTRÓNICO ZIMBRA APLICANDO TÉCNICA DE MACHINE LEARNIG DE CLASIFICACIÓN DE TEXTO PARA UN GAD MUNICIPAL”

Por lo antes descrito se le concede al Sr. Bryan Soria Méndez la apertura para realizar el trabajo de titulación en el Departamento correspondiente.

Particular que comunico a usted para los fines a seguir.

Atentamente.

C.c. Archivo.-
jrm.-

Anexo 2. Técnica de observación aplicada al panel de administración del servicio de Zimbra del administrador de servidores del departamento de TI.

UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN			
Realizado por:	Soria Méndez Bryan	**Nombre del reporte:**	Observación al panel de administración de Zimbra.
Fecha	10/04/2023		
Hora	08:30 AM – 10:30 AM		
Entendimiento del funcionamiento del panel de administración de Zimbra			
Metodología: Observación			
Objetivos de la fase: • Conocer el proceso que el administrador realiza para detectar anomalías en el servidor de correo electrónico.			
Resultados • El administrador hace uso del panel de administración del servicio de Zimbra constantemente. • Dentro del panel de administración se puede monitorear el flujo de trabajo del servicio. • El panel de administración proporciona información relevante como las sesiones activas y cola de correos. • Si un usuario ingresa un destinatario de manera incorrecta, se puede visualizar en el panel de administración en tiempo real. • Al finalizar el día, la cuenta Admin de Zimbra se reenvía un correo electrónico con el historial de uso del servicio, tales como los usuarios que más correos enviaron y recibieron. • El historial proporcionado por el Admin de Zimbra también muestra la cantidad de espacio en almacenamiento que los correos electrónicos han ocupado. • Observar el estado actual de los servicios que incorpora el servicio de Zimbra.			

- El administrador puede observar los correos entrantes, activos, retenidos y dañados.
- La cantidad aproximada de correos enviados y recibidos diariamente son de 200, contando los dominios internos y externos.

Anexo 3. Observación aplicada al servicio de correo electrónico Zimbra para conocer su funcionamiento.

UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN			
Realizado por:	Soria Méndez Bryan	**Nombre del reporte:**	Entendimiento de la organización
Fecha	26/06/2023		
Hora	08:30 AM – 12:30 AM		

Entendimiento de la organización

Metodología: Observación

Objetivos de la fase:

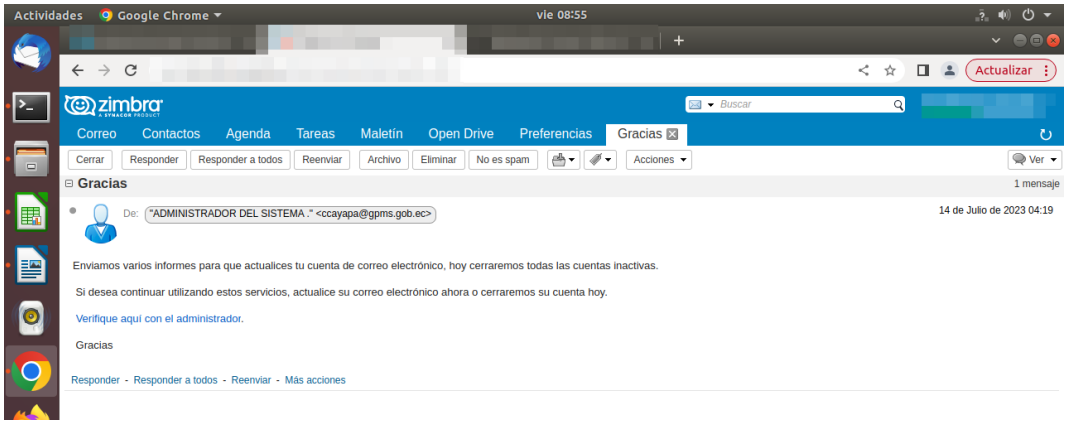
- Conocer el funcionamiento del servidor de correo electrónico.

Resultados

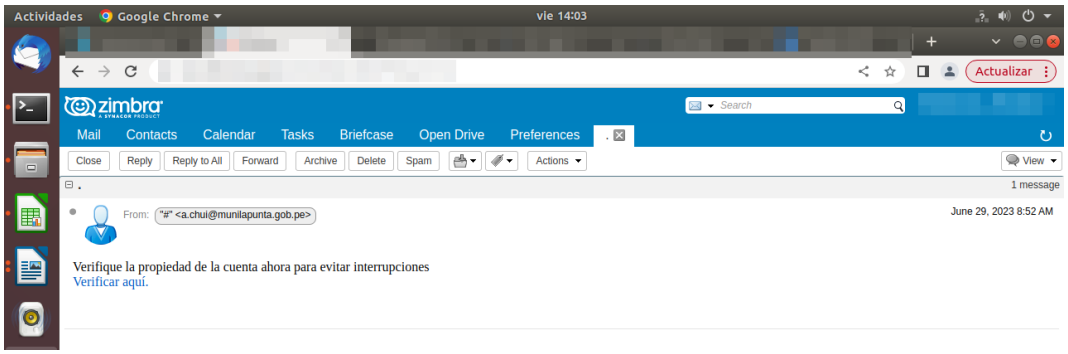
- Actualmente cuentan con un servidor de correo electrónico que brinda servicio a todo el GAD.
- El servidor de correo electrónico está virtualizado con Proxmox.
- Más de 300 usuarios poseen con una cuenta de correo electrónico.
- El servidor de correo electrónico cuenta con el servicio Zimbra
- Usan servidores propios para alojar el servicio de correo electrónico
- Todas las peticiones o solicitudes se realizan a través del correo electrónico.
- Los usuarios internos pueden enviar y recibir mensajes de dominios externos.
- Existen usuarios de correo que reciben mensajes categorizados como spam o de dudosa procedencia.

- Varios usuarios no cierran la sesión de correo cuando lo dejan de usar.
- Existe segmentación de la red.
- Cuentan con firewall que filtre información de red, pero no detecta si un correo electrónico es spam.
- Para que un usuario pueda reestablecer su contraseña, debe realizar un llamado al departamento de sistemas vía telefónica para pedir este proceso, o a su vez, acercarse al departamento
- Existen segmentaciones de red en el GAD, pero no en su totalidad.
- Los usuarios no tienen restricciones de accesos a páginas webs.
- Los usuarios pueden acceder a páginas webs que contengan código malicioso.
- No existe un medio de alerta hacia los usuarios o al administrador para detectar correos spam
- Para poder determinar si una cuenta de correo electrónico está enviando spam, se analizan los reportes diarios que se envían automáticamente al administrador por parte del servicio de Zimbra, pero estos reportes solo muestran la cantidad de correos electrónicos enviados y recibidos por persona.
- Al ser una entidad pública, la lista de los correos electrónicos del personal se encuentra disponible en la página web del GAD
- Los correos detectados como spam provienen de dominios externos del GAD.
- Los dominios externos detectados como spam suelen incluir alguna identificación de identificaciones públicas, tales como “gob.ec” e incluso dominios de países externos, como “pe” y “ec”, haciendo referencia a códigos postales de Perú y Ecuador respectivamente.

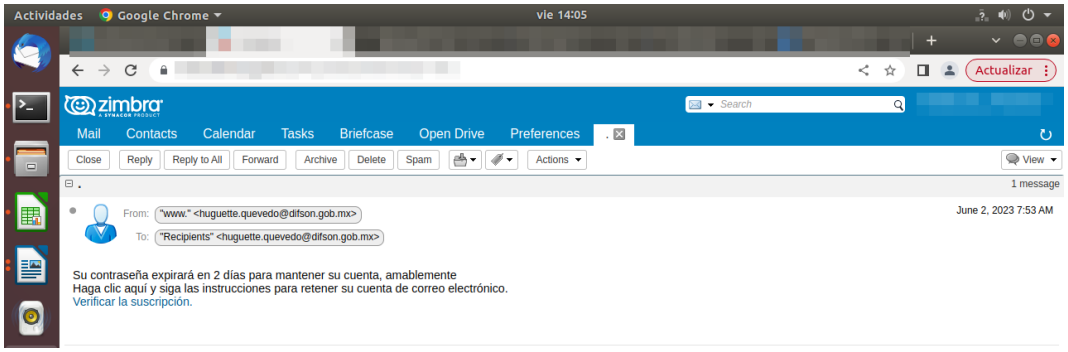
Anexo 4. Evidencia de existencia de correos spam en el servicio de Zimbra del GAD.



Anexo 5. Evidencia de existencia de correos spam en el servicio de Zimbra del GAD.



Anexo 6. Evidencia de existencia de correos spam en el servicio de Zimbra del GAD.



Anexo 7. Entrenamiento del algoritmo Random Forest.

```python
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split

# Definir el tamaño del chunk
chunk_size = 41000

# Cargar el archivo CSV en chunks
data_chunks = pd.read_csv('C:/1. final_dataset.csv', chunksize=chunk_size)

# Crear un vectorizador de texto para realizar la tokenización
vectorizer = CountVectorizer()

# Inicializar variables para almacenar resultados
y_true = []
y_pred = []

# Iterar sobre los chunks de datos
for chunk in data_chunks:
    # Dividir el conjunto de datos en características (X) y etiquetas (y)
    X = chunk['text']
    y = chunk['label']

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Ajustar el vectorizador al conjunto de entrenamiento actual
    vectorizer.fit(X_train)

    # Aplicar la tokenización a los datos de entrenamiento y prueba
    X_train_tokenized = vectorizer.transform(X_train)
    X_test_tokenized = vectorizer.transform(X_test)

    # Crear y entrenar el modelo RandomForest
    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train_tokenized, y_train)

    # Realizar predicciones en los datos de prueba
    y_pred_chunk = model.predict(X_test_tokenized)

    # Almacenar las etiquetas verdaderas y las predicciones del chunk actual
    y_true.extend(y_test)
    y_pred.extend(y_pred_chunk)

# Calcular el accuracy y el classification report para todos los chunks
accuracy = accuracy_score(y_true, y_pred)
report = classification_report(y_true, y_pred)

print("Accuracy:", accuracy)
print("Classification Report:")
print(report)
```

Anexo 8. Entrenamiento algoritmo Decision Tree.

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split

# Definir el tamaño del chunk
chunk_size = 10000

data_chunks = pd.read_csv('C:/1. final_dataset_tokenizacion.csv', chunksize=chunk_size)

# Crear un vectorizador de texto para realizar la tokenización
vectorizer = CountVectorizer()

# Inicializar variables para almacenar resultados
y_true = []
y_pred = []

# Iterar sobre los chunks de datos
for chunk in data_chunks:
    # Dividir el conjunto de datos en características (X) y etiquetas (y)
    X = chunk['text']
    y = chunk['label']

    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Ajustar el vectorizador al conjunto de entrenamiento actual
    vectorizer.fit(X_train)

    # Aplicar la tokenización a los datos de entrenamiento y prueba
    X_train_tokenized = vectorizer.transform(X_train)
    X_test_tokenized = vectorizer.transform(X_test)

    # Crear y entrenar el modelo Decision Tree
    model = DecisionTreeClassifier(random_state=42)
    model.fit(X_train_tokenized, y_train)

    # Realizar predicciones en los datos de prueba
    y_pred_chunk = model.predict(X_test_tokenized)

    # Almacenar las etiquetas verdaderas y las predicciones del chunk actual
    y_true.extend(y_test)
    y_pred.extend(y_pred_chunk)

# Calcular el accuracy y el classification report para todos los chunks
accuracy = accuracy_score(y_true, y_pred)
report = classification_report(y_true, y_pred)

print("Accuracy:", accuracy)
print("Classification Report:")
print(report)
```

Anexo 9. Entrenamiento algoritmo Naïve Bayes.

```
import pandas as pd
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split

chunk_size = 90000

data_chunks = pd.read_csv('C:/1. final_dataset_tokenizacion.csv', chunksize=chunk_size)
vectorizer = CountVectorizer()
y_true = []
y_pred = []

for chunk in data_chunks:
    X = chunk['text']
    y = chunk['label']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    vectorizer.fit(X_train)

    X_train_tokenized = vectorizer.transform(X_train)
    X_test_tokenized = vectorizer.transform(X_test)
    model = MultinomialNB()
    model.fit(X_train_tokenized, y_train)
    y_pred_chunk = model.predict(X_test_tokenized)
    y_true = y_test
    y_pred = y_pred_chunk

accuracy = accuracy_score(y_true, y_pred)
report = classification_report(y_true, y_pred)

print("Accuracy:", accuracy)
print("Classification Report:")
print(report)
```

Anexo 10. Entrevista realizada al administrador de los servidores del departamento de TI para la recolección de información de la fase identificar

UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN			
Desarrollo de un agente para la detección de spam en el servicio de correo electrónico Zimbra aplicando técnica de machine learning de clasificación de texto para un GAD municipal.			
Entrevistador:	Soria Méndez Bryan Andrés	**Nombre del reporte:**	Etapa de orientación Fase identificar
Fecha	28/06/2023		
Hora	09 AM – 10 AM		
Fase identificar			
Objetivos de la fase: • Conocer las medidas de seguridad que el GAD implementa para poder identificar los riesgos dentro del servidor del correo electrónico			

Procedimiento

Se realizó la entrevista al encargado del puesto de redes e infraestructura del departamento de TI.

1. **¿Existen procesos formales de gestión de riesgos implementado en el departamento de TI?**
   a. Totalmente en desacuerdo
   b. En desacuerdo
   c. Neutro
   d. De acuerdo
   e. Totalmente de acuerdo
2. **¿Existe un plan organizacional en el departamento que incluya la seguridad del servidor de correo electrónico?**
   a. Totalmente en desacuerdo
   b. En desacuerdo
   c. Neutro
   d. De acuerdo
   e. Totalmente de acuerdo
3. **¿El plan organizacional es actualizado regularmente para abordar nuevos riesgos?**
   a. Nunca
   b. Raramente
   c. A veces
   d. Frecuentemente
   e. Siempre
4. **¿Se han implementado filtros de correos spam en el servidor de correo electrónico?**
   a. Nunca
   b. Raramente
   c. A veces
   d. Frecuentemente
   e. Siempre
5. **¿Se han implementado filtros de malware en el servidor de correo electrónico?**
   a. Nunca
   b. Raramente
   c. A veces
   d. Frecuentemente
   e. Siempre
6. **¿Existe una clasificación de los activos de información que se envían y reciben a través del correo electrónico?**
   a. Nunca
   b. Raramente
   c. A veces
   d. Frecuentemente
   e. Siempre
7. **¿Se han identificado los activos críticos o sensibles que circulan en el correo electrónico?**
   a. Nunca

b. Raramente
c. A veces
d. Frecuentemente
e. Siempre

8. ¿Existe una política formal de seguridad cibernética sobre el servidor de correo electrónico?

a. Totalmente en desacuerdo
b. En desacuerdo
c. Neutro
d. De acuerdo
e. Totalmente de acuerdo

9. ¿Se comunican las políticas de seguridad en relación con el correo electrónico?

a. Nunca
b. Raramente
c. A veces
d. Frecuentemente
e. Siempre

10. ¿Se han asignado roles y responsabilidades con la gestión de riesgos del servidor de correo electrónico?

a. Totalmente en desacuerdo
b. En desacuerdo
c. Neutro
d. De acuerdo
e. Totalmente de acuerdo

Anexo 11. Entrevista realizada al administrador de los servidores del departamento de TI para la recolección de información de la fase proteger.

UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Desarrollo de un agente para la detección de spam en el servicio de correo electrónico Zimbra aplicando técnica de machine learning de clasificación de texto para un GAD municipal.

Entrevistador	Soria Méndez Bryan Andrés	**Nombre del reporte:**	Etapa de orientación Fase proteger
Fecha	28/06/2023		
Hora	09 AM – 10 AM		

Fase proteger

Objetivos de la fase:

- Conocer las medidas de seguridad que el GAD implementa para poder proteger el servidor del correo electrónico frente a los riesgos

Procedimiento

Se realizó la entrevista al encargado del puesto de redes e infraestructura del departamento de TI.

1. **¿Existen políticas documentadas para la gestión de credenciales de los usuarios del servidor de correo electrónico?**
    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Neutro
    d. De acuerdo
    e. Totalmente de acuerdo
2. **¿Las políticas establecen los pasos para emitir, administrar y revocar credenciales de los usuarios?**
    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Neutro
    d. De acuerdo
    e. Totalmente de acuerdo
3. **¿Se han establecido normas claras para el acceso físico al servidor de correo electrónico?**
    a. No se ha establecido
    b. Establecido, pero no se hace cumplir
    c. Establecido y ocasionalmente se hace cumplir
    d. Establecido y se hace cumplir regularmente
    e. Establecido y se hace cumplir rigurosamente
4. **¿Se han establecido normas claras para el acceso remoto al servidor de correo electrónico?**
    a. No se ha establecido
    b. Establecido, pero no se hace cumplir
    c. Establecido y ocasionalmente se hace cumplir
    d. Establecido y se hace cumplir regularmente
    e. Establecido y se hace cumplir rigurosamente
5. **¿Se aplican controles de acceso mínimo al servidor de correo electrónico tanto en términos físicos como remotos?**
    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Neutro
    d. De acuerdo
    e. Totalmente de acuerdo
6. **¿En qué medida el servidor de correo electrónico está configurado en una red segmentada?**
    a. Totalmente desprotegida, sin segmentación de red
    b. En su mayoría desprotegida, con mínima segmentación de red
    c. Algunas medidas de segmentación de red implementadas
    d. La mayoría de las medidas de segmentación de red implementadas
    e. Totalmente segmentada
7. **¿Se ha implementado un firewall para proteger el servidor de correo electrónico?**

a. No se ha implementado
b. Implementado, pero no se configura adecuadamente
c. Configurado de forma básica
d. Configurado de forma adecuada
e. Configurado de forma avanzada y optimizada

Anexo 12. Entrevista realizada al administrador de los servidores del departamento de TI para la recolección de información de la fase detectar.

UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Desarrollo de un agente para la detección de spam en el servicio de correo electrónico Zimbra aplicando técnica de machine learning de clasificación de texto para un GAD municipal.

Entrevistador	Soria Méndez Bryan Andrés	**Nombre del reporte:**	Etapa de orientación Fase detectar
Fecha	28/06/2023		
Hora	09 AM – 10 AM		

Fase detectar

Objetivos de la fase:

- Conocer las medidas de seguridad que el GAD implementa para poder detectar posibles riesgos dentro del servidor de correo electrónico

Procedimiento

Se realizó la entrevista al encargado del puesto de redes e infraestructura del departamento de TI.

1. **¿En qué medida se aplican herramientas de alertas en caso de incidentes relacionados con el correo electrónico?**
   a. No
   b. Limitado
   c. Moderado
   d. Efectivo
   e. Avanzado
2. **¿En qué medida se aplican sistemas de monitoreo de red para detectar posibles eventos de ciberseguridad relacionados con el correo electrónico?**
   a. No
   b. Limitado
   c. Moderado
   d. Efectivo
   e. Avanzado

3. ¿Se aplican sistemas de registro de actividad para la identificación de comportamientos inusuales relacionados con el correo electrónico?

a. No
b. Limitado
c. Moderado
d. Efectivo
e. Avanzado

4. ¿Se aplican sistemas para la detección de código malicioso en correos electrónicos?

a. No
b. Limitado
c. Moderado
d. Efectivo
e. Avanzado

Anexo 13. Entrevista realizada al administrador de los servidores del departamento de TI para la recolección de información de la fase responder.

UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Desarrollo de un agente para la detección de spam en el servicio de correo electrónico Zimbra aplicando técnica de machine learning de clasificación de texto para un GAD municipal.

Entrevistador	Soria Méndez Bryan Andrés	**Nombre del reporte:**	Etapa de orientación Fase responder
Fecha	28/06/2023		
Hora	09 AM – 10 AM		

Fase Responder

Objetivos de la fase:

- Conocer las medidas de seguridad que el GAD implementa para poder responder frente a posibles riesgos dentro del servidor de correo electrónico

Procedimiento

Se realizó la entrevista al encargado del puesto de redes e infraestructura del departamento de TI.

1. ¿Se aplican respuestas a incidentes durante o después del suceso relacionado con el correo electrónico?

a. No
b. Pocas medidas
c. Medidas adecuadas

d. Medidas eficaces
e. Medidas avanzadas

2. **¿**Se analizan las notificaciones de los sistemas de detección relacionadas con el correo electrónico**?**

a. No
b. Limitado
c. Moderado
d. Efectivo
e. Avanzado

3. **¿**Se comprende el impacto de tener spam en los correos electrónicos**?**

a. No
b. Comprendido de manera limitada
c. Comprendido de manera adecuada
d. Comprendido de manera efectiva
e. Comprendido completamente

4. ¿Se toman medidas para evitar la propagación del incidente relacionado con el correo electrónico?

a. No
b. Limitado
c. Moderado
d. Efectivo
e. Avanzado

5. ¿Se toman acciones inmediatas para mitigar los efectos del incidente relacionado con el correo electrónico?

a. No
b. Limitado
c. Moderado
d. Efectivo
e. Avanzado

6. ¿Los incidentes ocurridos son mitigados o documentados como riesgo aceptado relacionado con el correo electrónico?

a. No
b. Limitado
c. Moderado
d. Efectivo
e. Avanzado

Anexo 14. Entrevista realizada al administrador de los servidores del departamento de TI para la recolección de información de la fase recuperar.

UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Desarrollo de un agente para la detección de spam en el servicio de correo electrónico Zimbra aplicando técnica de machine learning de clasificación de texto para un GAD municipal.

Entrevistador	Soria Méndez Bryan Andrés	**Nombre del reporte:**	Etapa de orientación Fase recuperar
Fecha	28/06/2023		
Hora	09 AM – 10 AM		

Fase Recuperar

Objetivos de la fase:

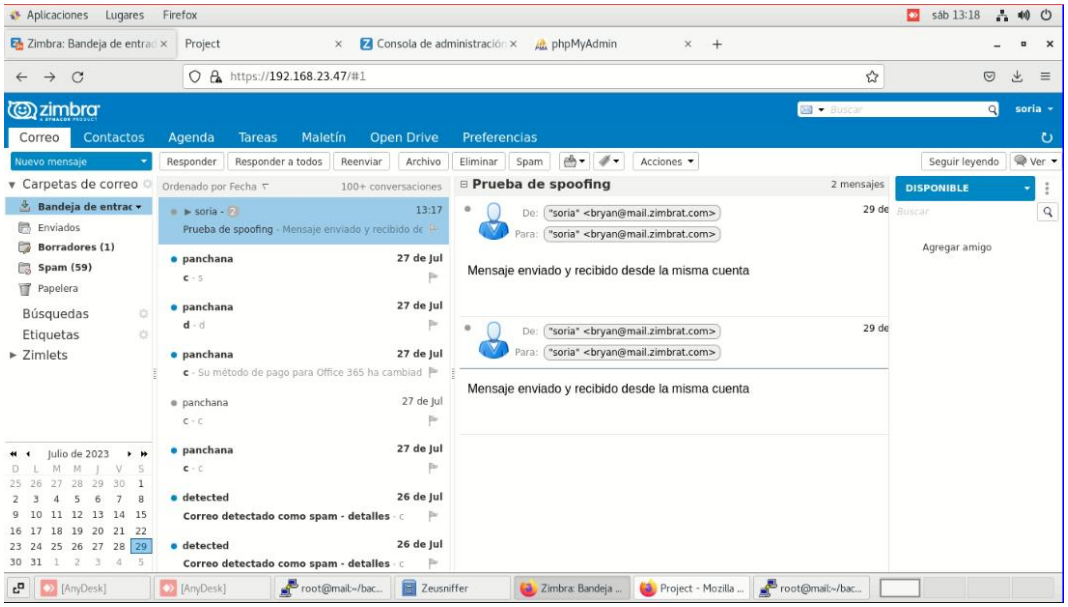
- Conocer las medidas de seguridad que el GAD implementa para poder recuperar la información frente a posibles riesgos dentro del servidor de correo electrónico

Procedimiento

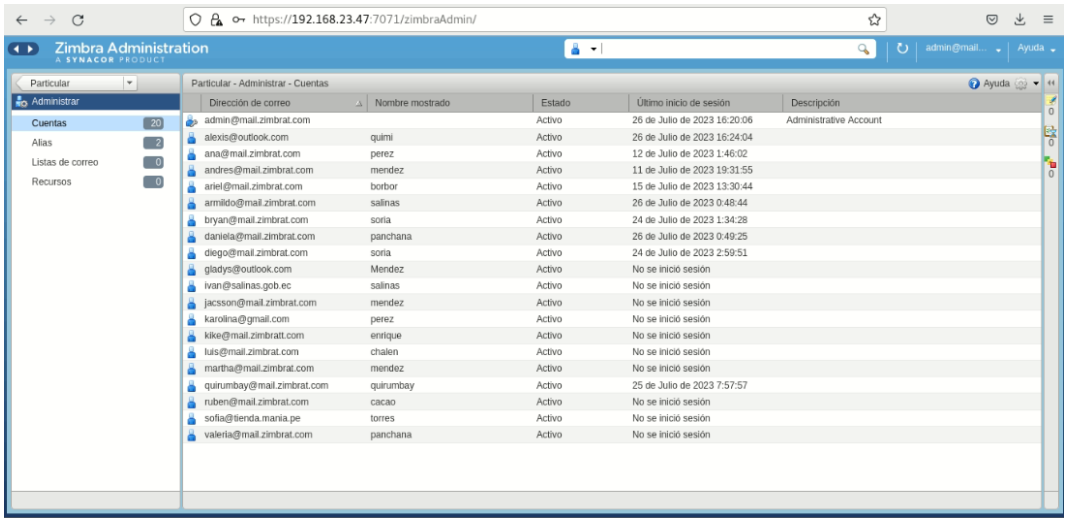
Se realizó la entrevista al encargado del puesto de redes e infraestructura del departamento de TI.

1. ¿Existen un proceso formal de gestión de riesgos implementado en el departamento de TI?
    a. No
    b. Limitado
    c. Moderado
    d. Efectivo
    e. Avanzado
2. ¿Se realizan copias de seguridad regulares de los datos y sistemas críticos?
    a) Totalmente en desacuerdo
    b) En desacuerdo
    c) Neutro
    d) De acuerdo
    e) Totalmente de acuerdo

Anexo 15. Evidencia de remitente y receptor iguales dentro del servicio de Zimbra en un entorno controlado.



Anexo 16. Usuarios registrados en el servicio de Zimbra en un entorno controlado.



Anexo 17. Funciones del marco de ciberseguridad de la NIST y sus funciones.

Identificador único de función	Función	Identificador único de categoría	Categoría
ID	Identificar	ID.AM	Gestión de activos
		ID.BE	Entorno empresarial
		ID.GV	Gobernanza
		ID.RA	Evaluación de riesgos
		ID.RM	Estrategia de gestión de riesgos

		ID.SC	Gestión del riesgo de la cadena de suministro
PR	Proteger	PR.AC	Gestión de identidad y control de acceso
		PR.AT	Conciencia y capacitación
		PR.DS	Seguridad de datos
		PR.IP	Procesos y procedimientos de protección de la información
		PR.MA	Mantenimiento
		PR.PT	Tecnología protectora
DE	Detectar	DE.AE	Anomalías y eventos
		DE.CM	Vigilancia continua de seguridad
		DE.DP	Procesos de detección
RS	Responder	RS.RP	Planificación de respuesta
		RS.CO	Comunicaciones
		RS.AN	Análisis
		RS.MI	Mitigación
		RS.IM	Mejoras
RC	Recuperar	RC.RP	Planificación de recuperación
		RC.IM	Mejoras
		RC.CO	Comunicaciones

Anexo 18. Niveles de implementación del marco CSF de la NIST [10].

Identificador	Nivel	Descripción
1	Parcial	La organización no cuenta con un proceso formalizado y todas las gestiones de riesgos se los hace de manera ad hoc y reactiva
2	Informado	Las prácticas de gestión de riesgos han sido aprobadas por el personal administrativo, pero no está formalizado a nivel organizacional
3	Repetible	Las prácticas para la gestión de riesgos se aprueban formalmente y se expresan como políticas, actualizandose perídicamente dependiendo de las amenazas encontradas.
4	Adaptativo	Representa un alto nivel de madurez en la organización sobre la gestión de riesgos, adaptando las prácticas de seguridad cibernética basandose en actividades previas, ajustando esta gestion a cada servicio que cuenten

Anexo 19. Identificación de los niveles de implementación actuales de cada función del núcleo de la metodología CSF.

Función	Categoría	Subcategoría	Estado actual	
			Nivel	**Justificación**
IDENTIFICAR (ID)	**Gestión de activos (ID.AM):**	**ID.AM-2:** Las plataformas de software y las aplicaciones dentro de la organización están inventariadas.	1 Parcial	• Actualmente el GAD no realiza un registro de las actualizaciones de seguridad que se realizan en el servidor de correo electrónico. • Todas las actualizaciones se las realizan normalmente después de los horarios laborales. • Actualmente el servidor de correo electrónico se encuentra operativo en el sistema operativo Centos7, en el cual para el año 2024 deje de recibir soporte por la comunidad.
	Resumen de la categoría "ID-AM: Gestión de activos"		**%Logro 25,00%**	
	Evaluación de riesgos (ID.RA)	**ID.RA-1**: Se identifican y se documentan las vulnerabilidades de los activos.	1 Parcial	No se identifican las vulnerabilidades que pueden sufrir los activos, pero si se es consciente de las amenazas que puede ocurrir de manera general
	Resumen de la categoría "ID-RA: Evaluación de riesgos"		**%Logro 25,00%**	
	Estrategia de gestión de riesgos (ID.RM)	**ID.RM-1:** Los actores de la organización establecen, gestionan y acuerdan los procesos de gestión de riesgos.	1 Parcial	Actualmente no poseen una gestión de riesgos enfocada al servicio de correo electrónico

	Resumen de la categoría "ID-RM: Estrategia de gestión de riesgos"		%Logro 25,00%	
PROTEGER (PR)	**Gestión de identidad, autenticación y control de acceso (PR.AC)**	**PR.AC-1:** Las identidades y credenciales se emiten, se administran, se verifican, se revocan y se auditan para los dispositivos, usuarios y procesos autorizados.	2 Riesgo Informado	• Las credenciales se establecen por el administrador, añadiendo parámetros a los nombres de usuario y contraseñas. • El administrador tiene anotado las claves de acceso a la cuenta admin de Zimbra anotadas en un papel. • Cuando los usuarios se olvidan las contraseñas acueden al administrado para reestablecerlas y las llevan anotados en un papel o les toman una fotografía. • En el Anexo 18 se puede identificar que la dirección de correo detected proveniente del agente ha enviado una alerta, pero esta cuenta no ha sido registrada, por lo que puede ser un indicio de que los atacantes usan el dominio del GAD para poder enviar correos electrónicos sin necesidad de que su correo electrónico esté registrado.
		PR.AC-2: Se gestiona y se protege el acceso físico a los activos.	2 Riesgo Informado	• Actualmente establecen seguridad para entrar al centro de datos, solo se admiten dos personas a la vez, pero la norma no es formal. • Hay veces en que la que ingresan más de dos personas al centro de datos.
		PR.AC-4: Se gestionan los permisos y autorizaciones de acceso con incorporación de los principios de menor privilegio y separación de funciones.	2 Riesgo Informado	• El administrador define los permisos y privilegios de los usuarios existente en el servidor de correo electrónico. • El administrador realiza todas las acciones desde el usuario root.

	Resumen de la categoría "PR-AC: Gestión de identidad, autenticación y control de acceso"		**%Logro 50%**	
	Seguridad de los datos (PR.DS).	**PR.DS-2:** Los datos en tránsito están protegidos.	2 Riesgo Informado	• Los datos están protegidos por los protocolos que aplica el servidor Zimbra, añadiendo la seguridad de cifrado que tiene el dominio web de GAD. • Se puede navegar sin limitaciones en las redes de cada departamento, es decir, no existe bloqueos de páginas webs y contenido en general.
		PR.DS-4: Se mantiene una capacidad adecuada para asegurar la disponibilidad.	2 Riesgo Informado	Actualmente no se aplican métodos de disponibilidad
		PR.DS-5: Se implementan protecciones contra las filtraciones de datos.	1 Parcial	• No se implementan sistemas para filtrar correos electrónicos normales y spam. • Los correos electrónicos clasificados como spam llegan a la bandeja de entrada del usuario.
	Resumen de la categoría "PR-DS: Seguridad de los datos"		**%Logro 41,67%**	
DETECTAR (DE)	**Anomalías y Eventos (DE.AE):**	**DE.AE-5:** Se establecen umbrales de alerta de incidentes.	1 Parcial	No se establecen alertas para identificar la presencia de correos spam.
	Resumen de la categoría "DE-AE: Anomalías y Eventos"		**%Logro 25,00%**	

RESPONDER (RS)	**Análisis (RS.AN)**	**RS.AN-1:** Se investigan las notificaciones de los sistemas de detección.	1 Parcial	No se aplican alertas relacionadas a la detección de correos spam
		RS.AN-2: Se comprende el impacto del incidente.	2 Riesgo Informado	El administrador del servidor comprende el riesgo que conlleva el spam en el correo electrónico
	Resumen de la categoría "RS-AN: Análisis"		**%Logro 37,50%**	
	Mitigación (RS.MI)	**RS.MI-1:** Los incidentes son contenidos.	1 Parcial	No existe un agente que remueva los correos detectados como spam de la bandeja de entrada del usuario
		RS.MI-2: Los incidentes son mitigados.	1 Parcial	No existen estadísticas de correos detectados como spam para que el administrador pueda tomar decisiones tales como aplicar filtros o bloquear direcciones de correos electrónicos del servicio de Zimbra
	Resumen de la categoría "RS-MI: Mitigación"		**%Logro 25,00%**	
RECUPERAR (RC)	**Planificación de la recuperación (RC.RP)**	**RC.RP-1:** El plan de recuperación se ejecuta durante o después de un incidente de seguridad cibernética.	1 Riesgo Parcial	Aunque no se ha sufrido una pérdida de información, el departamento depende de las copias de seguridad establecidas diariamente de manera local, sin embargo, no emplean un plan formal.
	Resumen de la categoría "RC-RP: Planificación de la recuperación"		**%Logro 25,00%**	

Anexo 20. Análisis de riesgo enfocada al servicio de correo electrónico.

			Impacto	Probabilidad	Medición	
Función	**Identificador**	**Riesgo servidor de correo electrónico**	Valor	Valor	Total	**Medición**
Identificar	ID.AM2	Falta de control de cambios y acciones	5	1	5	Medio
		Vulneración del sistema operativo Centos7 mediante exploits	5	1	5	Medio
	ID.RA1	Dificultad para implementar medidas de seguridad	4	1	4	Medio
	ID.RM1	Poca respuesta para identificar, evaluar y minimizar un riesgo	5	1	5	Medio
Proteger	PR.AC1	Phishing	5	1	5	Medio
		Acceso no autorizado a la cuenta admin de Zimbra por robo de credenciales	5	1	5	Medio
		Divulgación de contraseñas de los usuarios	5	2	10	Alto
		Poco control para mitigar el envío de spam de cuentas que no están registradas en el servidor Zimbra	3	5	15	Muy Alto
	PR.AC2	Accesos no autorizados al centro de datos	4	2	8	Medio
	PR.AC4	Uso inapropiado del usuario root	4	2	8	Medio
	PR.DS2	Acceso a páginas que recopile información personal	4	2	8	Muy alto
		Acceso a sitios que contengan phishing, spam y código malicioso	4	2	8	Medio

	PR.DS4	Denegación de servicio	5	1	5	Alto
	PR.DS5	Poca capacidad del administrador al detectar direcciones de correo electrónico que propaguen spam	4	3	12	Alto
		Distribución de phishing	4	3	12	Alto
		Distribución de código malicioso	4	1	4	Medio
Detectar	DE.AE5	Detección tardía de correos electrónicos con spam	4	3	12	Alto
Responder	RS.AN1	Falta de detección de correos spam	4	3	12	Alto
	RS.MI1	Falta de contención de correos spam	4	3	12	Alto
		Expansión no controlada de spam	3	4	12	Alto
		Permanencia de correos electrónicos spam en la bandeja de entrada de los usuarios	3	4	12	Alto
	RS.MI2	Poca capacidad del administrador para aplicar filtros de direcciones de correos electrónicos que envíen spam	4	3	12	Alto
Recuperar	RC.RP1	Falta de coordinación al recuperar los datos por no existir un plan formal de recuperación de servicio	5	1	5	Medio
		Dependencia de las copias de seguridad locales	5	1	5	Medio
		Interrupción prolongada del servicio	5	1	5	Medio

Anexo 21. Plan de acción enfocada al servicio de correo electrónico.

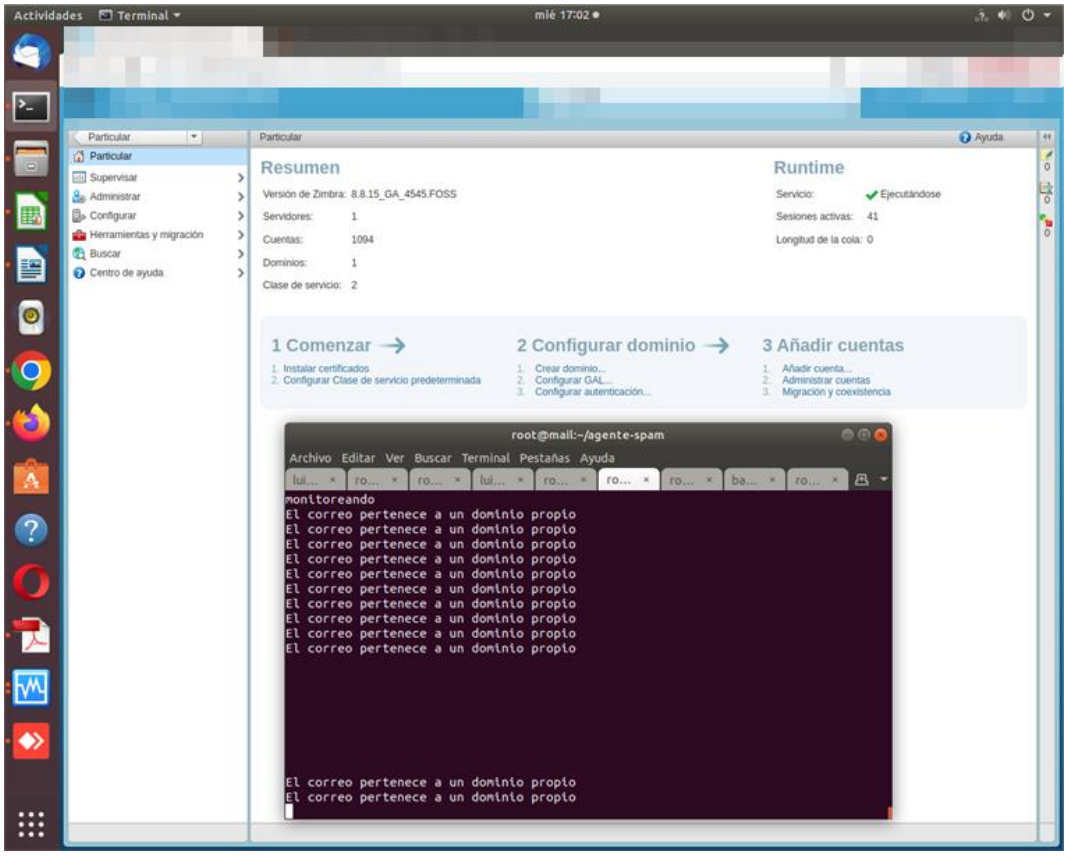
Función	Identificador	Plan de acción del servidor de correo electrónico
Identificar	ID.AM2	• Mantener el sistema operativo actualizado. • Realizar un inventario de las actualizaciones realizadas. • Migrar el servicio de Zimbra desde el sistema operativo Centos7.
	ID.RA1	• Asignar responsabilidades para la gestión de vulnerabilidades. • Realizar evaluaciones regulares sobre vulnerabilidades en el servidor de correo. • Realizar documentación de las vulnerabilidades identificadas. • Implementar medidas de seguridad con base a las vulnerabilidades encontradas.
	ID.RM1	• Desarrollar un plan de gestión de riesgos basado en el servicio de correo electrónico.
Proteger	PR.AC1	• Establecer procesos formales para la emisión de cuentas de correo electrónico. • Establecer procesos formales para la revocación de cuentas de correo electrónico. • Establecer procesos formales para la generación y entrega de contraseñas para cada usuario. • Verificación de cuentas inactivos. • Desactivación de las cuentas inactivas.
	PR.AC2	• Implementar medidas de seguridad físicas para el acceso al centro de datos.

		• Implementar sistemas de monitoreo y acceso al centro de datos. • Establecer sistemas de registro al centro de datos.
	PR.AC4	• Asignación mínima a los usuarios existentes en el servidor de correo electrónico. • Desactivar el usuario root del servidor de correo electrónico. • Crear un usuario con privilegios similares al usuario root. • Desactivar usuarios inactivos.
	PR.DS2	• Implementación de reglas en el firewall para el acceso restringido a páginas. • Implementación de sistemas de detección de código malicioso. • Mantener los equipos informáticos de los usuarios actualizados.
	PR.DS4	• Implementar métodos de redundancia en el servidor de correo electrónico • Realizar copias de seguridad diarias del servidor de correo electrónico • Implementar reglas en el firewall del servidor de correo electrónico
	PR.DS5	• Implementar sistemas de detección de spam. • Implementar sistemas de alertas de detección de spam. • Capacitar a los usuarios del GAD sobre los diferentes de tipo de spam y sus riesgos. • Implementar antivirus en los equipos informáticos de los usuarios para prevenir archivos con código malicioso.
Detectar	DE.AE5	• Implementar sistemas de alertas de spam a los usuarios y administrador.

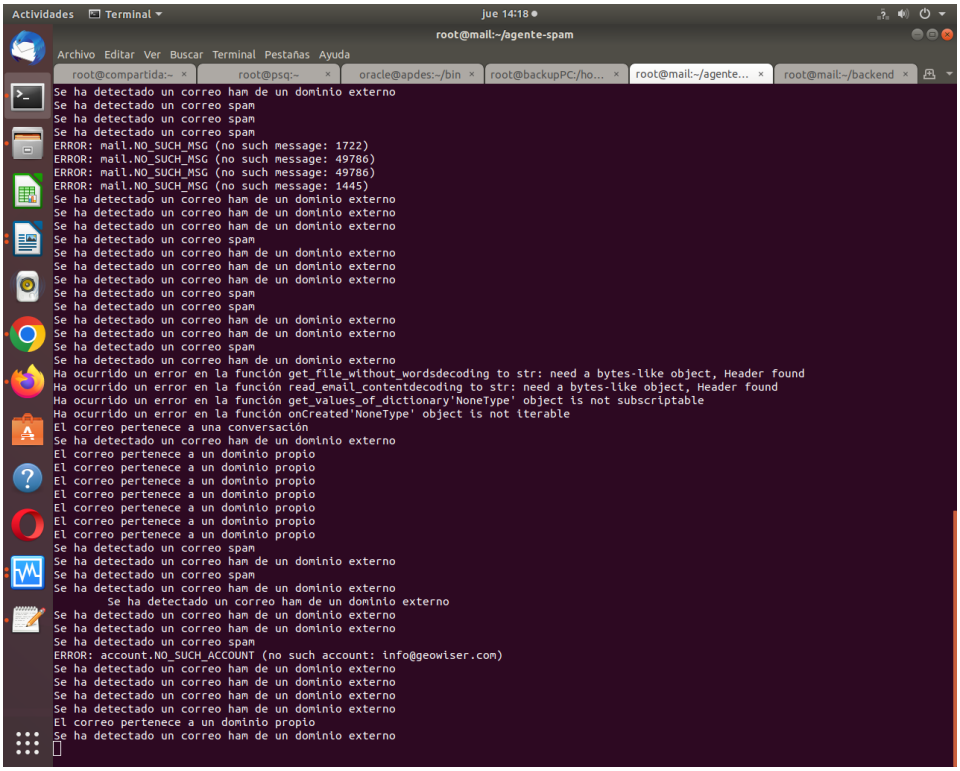
		• Implementar sistemas que permitan reportar la existencia de correos detectados como spam
Responder	RS.AN1	• Implementar soluciones con base a las alertas emitidas de los sistemas de detección de spam tales como bloquear las cuentas identificadas como emisoras de spam o rechazar la recepción de correos electrónicos a través de dominios.
	RS.MI1	• Implementar filtrado de correos electrónicos proveniente de usuarios detectados como emisores de spam.
	RS.MI2	• Emplear sistemas que redireccionen el correo detectado como spam a la carpeta correspondiente. • Emplear sistemas que envíen alertas a los usuarios de un correo detectado como spam y que este pueda reportarlo.
Recuperar	RC.RP1	• Establecer planes formales para la recuperación de datos • Establecer copias de seguridad locales y externas • Implementar métodos de alta disponibilidad en el servidor de correo electrónico

Anexo 22. Instalación del agente detector de spam con la asistencia del administrador del servidor.

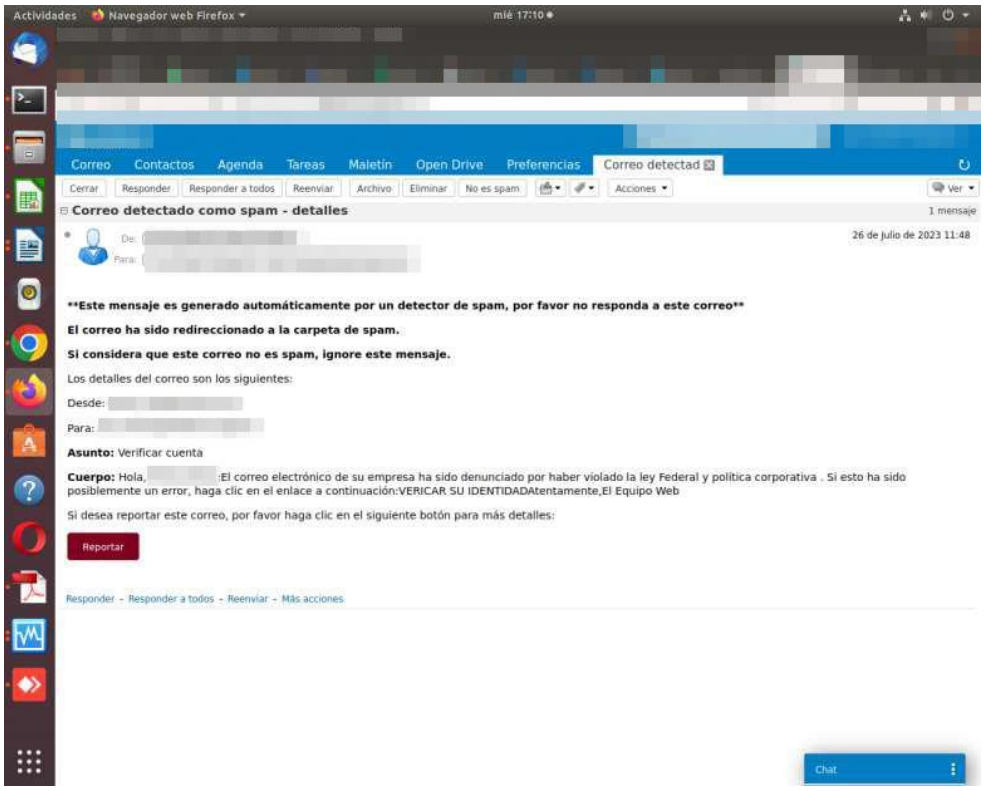
Anexo 23. Ejecución del agente en el servidor de correo electrónico del GAD municipal.



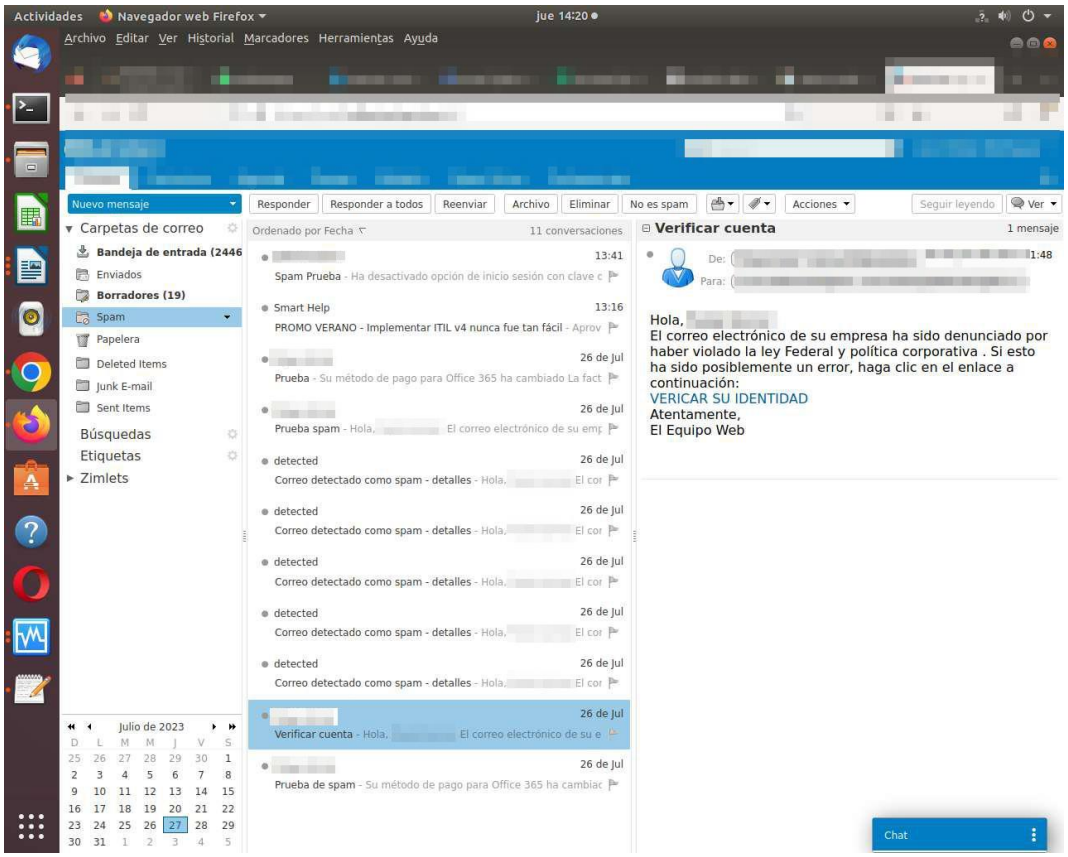
Anexo 24. Detección de spam dentro del servidor de correo electrónico del GAD municipal.



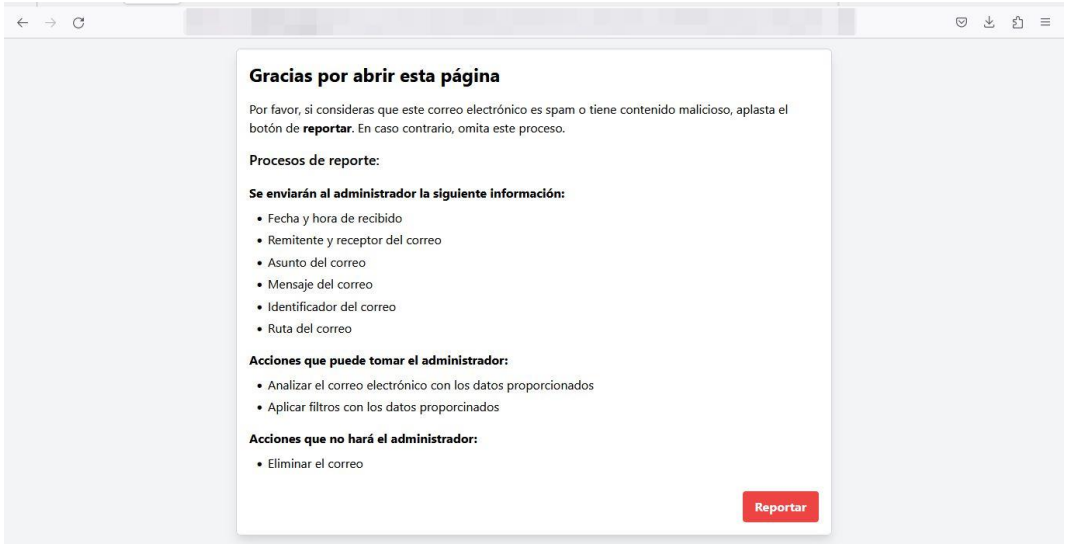
Anexo 25. Alerta enviada desde el agente hacia el usuario que recibió spam.



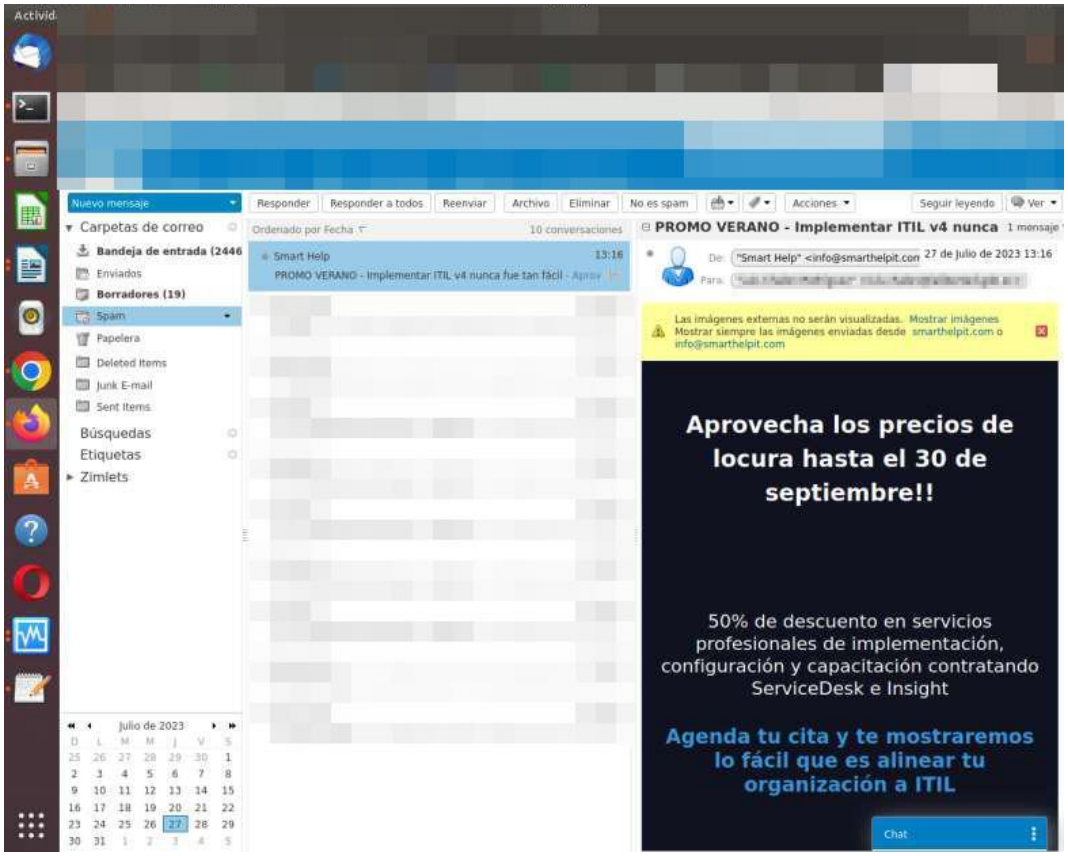
Anexo 26. Correo detectado como no deseado removido hacia la carpeta spam



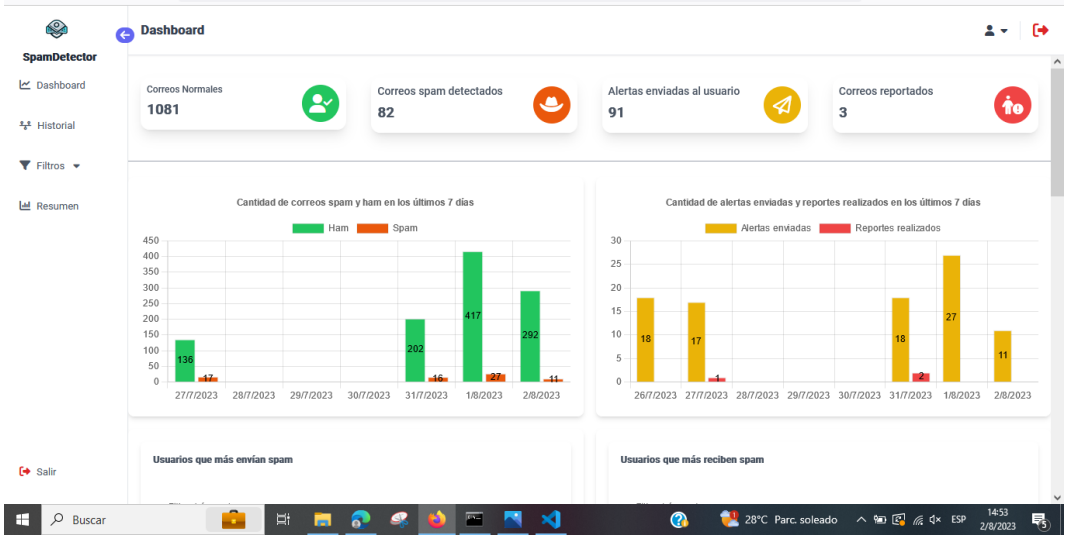
Anexo 27. Vista para reportar el correo detectado como spam.



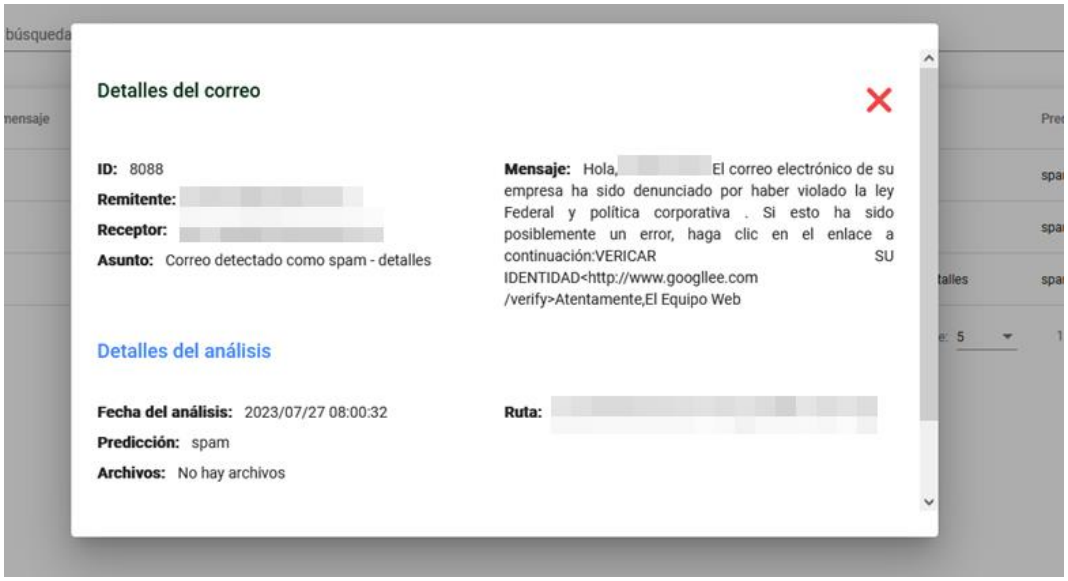
Anexo 28. Evidencia de que el agente removió el correo spam hacia la carpeta correspondiente.



Anexo 29. Gráficas con los datos almacenados por el agente detector de spam implementado en el servidor del GAD.

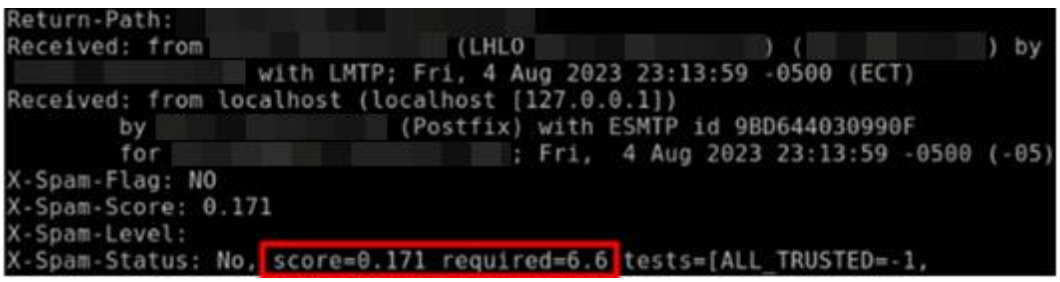


Anexo 30. Detalles del correo reportado como spam mediante el consentimiento del usuario.

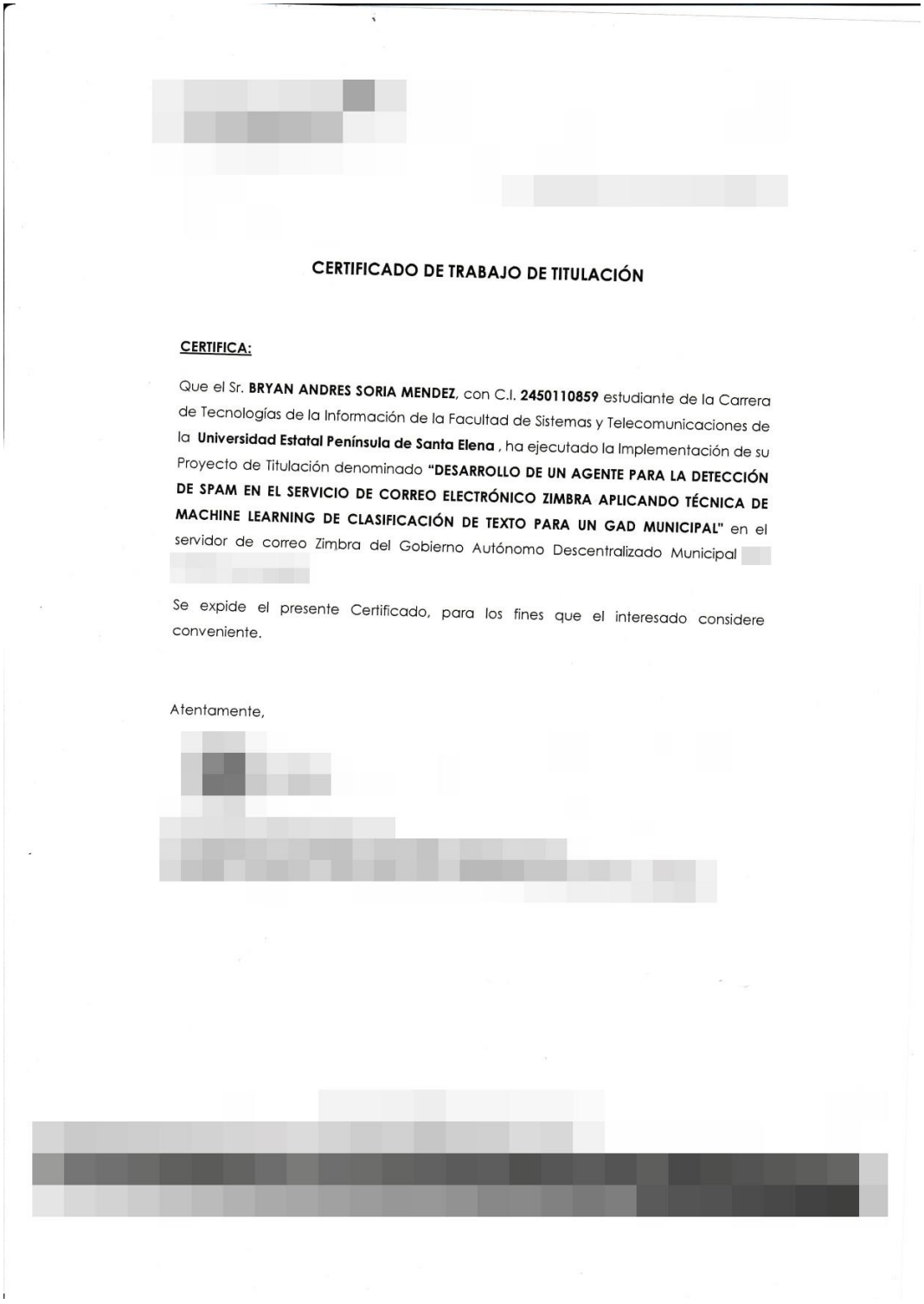


Anexo 31. Envío de correo spam en un entorno controlado para evaluar la eficiencia del sistema antispam de Zimbra.

Anexo 32. Calificación de spam del sistema antispam de Zimbra.



Anexo 33. Carta de implementación exitosa del agente detector de spam en el GAD municipal.



CERTIFICADO DE TRABAJO DE TITULACIÓN

CERTIFICA:

Que el Sr. **BRYAN ANDRES SORIA MENDEZ**, con C.I. **2450110859** estudiante de la Carrera de Tecnologías de la Información de la Facultad de Sistemas y Telecomunicaciones de la **Universidad Estatal Península de Santa Elena** , ha ejecutado la Implementación de su Proyecto de Titulación denominado **"DESARROLLO DE UN AGENTE PARA LA DETECCIÓN DE SPAM EN EL SERVICIO DE CORREO ELECTRÓNICO ZIMBRA APLICANDO TÉCNICA DE MACHINE LEARNING DE CLASIFICACIÓN DE TEXTO PARA UN GAD MUNICIPAL"** en el servidor de correo Zimbra del Gobierno Autónomo Descentralizado Municipal

Se expide el presente Certificado, para los fines que el interesado considere conveniente.

Atentamente,