



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Sistemas de prevención de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning: Caso práctico ataques Phishing.

Armas Ruales Jorge Andres y Simbaña Shuguli, Anthony Alexander

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Trabajo de integración curricular, previo a la obtención del título de Ingeniero de
Software

Ing. Corral Diaz, María Alexandra. Msc

21 de Agosto del 2023

Latacunga



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Reporte de Verificación de contenido



Tesis- Sistemas de prevención de intr...

Scan details

Scan time:
August 24th, 2023 at 17:9 UTC

Total Pages:
103

Total Words:
25516

Plagiarism Detection



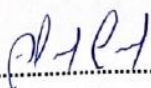
Types of plagiarism		Words
Identical	0.8%	193
Minor Changes	0.1%	25
Paraphrased	4.4%	1113
Omitted Words	9.8%	2497

AI Content Detection



Text coverage
 AI text
 Human text

i Alerts: (1)


.....

Ing. Corral Díaz, María Alexandra, Msc

C. C.: 0501970487

Certificación



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Certificación

Certifico que el trabajo de integración curricular: **“Sistemas de prevención de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning: Caso práctico ataques Phishing”** fue realizado por los señores **Armas Ruales, Jorge Andres y Simbaña Shuguli, Anthony Alexander**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 21 de agosto de 2023

Una firma manuscrita en tinta azul que parece decir 'MAD'.

Ing. Corral Díaz, María Alexandra. Msc

C. C: 0501970487

Responsabilidad de Auditoria.



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Responsabilidad de Auditoria

Nosotros, **Armas Ruales, Jorge Andres y Simbaña Shuguli, Anthony Alexander**, con cédulas de ciudadanía N° ° 0504046913 y 1719368654, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **"Sistemas de prevención de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning: Caso práctico ataques Phishing"**, es nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas

Latacunga, 21 de agosto de 2023

Firma manuscrita en tinta azul de Jorge Andres Armas Ruales.

Armas Ruales, Jorge Andres

C .C: 0505046913

Firma manuscrita en tinta azul de Simbaña Shuguli, Anthony Alexander.

Simbaña Shuguli, Anthony Alexander

C. C: 1719368654

Autorización de Publicación



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Autorización de Publicación

Nosotros, **Armas Ruales, Jorge Andres y Simbaña Shuguli, Anthony Alexander**, con cédulas de ciudadanía N° 0504046913 y 1719368654, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **“Sistemas de prevención de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning: Caso práctico ataques Phishing”**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 21 de agosto de 2023

Firma manuscrita en azul de Jorge Andres Armas Ruales.

Armas Ruales, Jorge Andres

C .C: 0505046913

Firma manuscrita en azul de Anthony Alexander Simbaña Shuguli.

Simbaña Shuguli, Anthony Alexander

C. C: 1719368654

Dedicatoria

Dedico este trabajo a mi madre, por su amor y apoyo constante en toda mi vida universitaria, que me ha guiado en cada paso de mi educación, a mi hermano que siempre confió en mí, en todo momento, ha estado ahí para apoyarme en lo que necesitara, a mi novia que ha sido mi luz, en la oscuridad y mi inspiración al momento de hacer este trabajo. Cuando los tiempos son oscuros, fue mi luz, cuando estuve perdido, fue la estrella que me guío.

También dedico este trabajo a mi familia y sobrinos que es una gran inspiración s sobrinos por las risas que me dieron durante todo este proceso.

Por último, quiero dedicar este trabajo a la Ing. María Alexandra Corral Diaz Msc, cuyo conocimiento me ayudo a entender cómo hacer, su orientación en el desarrollo de este trabajo. Aprecio mucho su ayuda y la oportunidad de haber sido mi guía invaluable.

Jorge Andres Armas Ruales

Ecuador, agosto de 20223

Dedicatoria

En el presente proyecto de titulación realizado con dedicación y esfuerzo, se lo dedico con mucho cariño y amor a mi familia, en especial a mis padres Jorge y Elizabeth, fueron un apoyo incondicional en el transcurso de mi vida académica, me enseñaron buenos valores para ser una persona que camine por el camino correcto, el sacrificio que cada uno de ellos realizo a lo largo de estos años.

También quiero dedicar este trabajo a mis tíos Ximena, Cesar y mi abuelita María los cuales son una parte fundamental en mi vida, me dieron la mano cuando más lo necesitaba, espero tener el apoyo de ellos por el resto de mi vida.

Finalmente, y no menos importantes dedico también a mis hermanos los cuales son una pieza que completa mi vida, me han apoyado de una manera muy grata y sin querer algo a cambio

Anthony Alexander Simbaña Shuguli

Ecuador, agosto 2023

Agradecimiento

Quiero agradecer a dios con por la guía que ha dado y darme la fortaleza para seguir adelante, también expresar mi más profundo agradecimiento a mi madre, por su amor incondicional y su apoyo constante, quien me brindo respaldo emocional y financiero. También agradecer a mi familia, mis amigos y a mi querida novia que ha sido mi gran apoyo en todo momento, a pesar de que hubo veces que me hubiera rendido si no hubiera sido por ella y su fe en mi durante este viaje, que ha sido toda una experiencia.

También quiero agradecer a mi hermano por su apoyo y preocupación por mi bienestar emocional y físico durante el transcurso de mi vida académica, apoyo y por creer en mi en todo momento.

Por último, quiero dedicar este trabajo a la Ing. María Alexandra Corral Diaz Msc, cuyo conocimiento me ayudo a entender cómo hacer, su orientación en el desarrollo de este trabajo. Aprecio mucho su ayuda y la oportunidad de haber sido mi guía invaluable.

Jorge Andres Armas Ruales

Ecuador, agosto de 20223

Agradecimiento

En primera instancia agradezco a Dios por haberme guiado por el camino correcto con el único fin de cumplir mis sueños, el cual estoy logrando a lo largo de estos años. Agradezco a mis padres por nunca rendirse con respecto a mis estudios, dándome el apoyo que necesitaba tanto económicamente, como sentimentalmente. También quiero agradecer a mis tíos Ximena, Cesar y mi abuelita María los años que me han apoyado de manera sentimental y económica, también quiero agradecer a mis hermanos que son importantes en mi vida por el apoyo incondicional.

Quiero agradecer a mi mejor amiga por brindarme esa amistad a lo largo de estos 10 años que la conozco, apoyándome virtualmente en no rendirme y ser una persona la cual me ha escuchado y aconsejado.

Agradezco a mis amigos Adrián, Dennis, Jorge que fueron un apoyo incondicional a lo largo de estos 4 años en los cuales hemos vivido experiencias que recordare con una sonrisa, juntos nos encontramos y finalmente juntos cumplimos el sueño de graduarnos como ingenieros, espero poder seguir llevándonos como siempre lo hemos hecho y les deseo suerte a cada uno en su vida profesional, aunque tomemos caminos distintos, siempre estaremos para apoyarnos.

Finalmente agradezco a mi tutora de tesis Ing. María Alexandra Corral Diaz, por las enseñanzas brindadas, por dirigirnos en el proceso de titulación y ser una persona de ejemplo en la que podemos confiar.

Anthony Alexander Simbaña Shuguli

Ecuador, agosto 2023

ÍNDICE DE CONTENIDO

Carátula.....	1
Reporte de Verificación de contenido	2
Certificación	3
Responsabilidad de auditoria.	4
Autorización de publicación	5
Dedicatoria	6
Dedicatoria	7
Agradecimiento.....	8
Agradecimiento.....	9
Indice de contenido	10
Índice de tablas	13
Índice de figuras	15
Resumen.....	17
Abstract	18
Capítulo I: Introducción.....	19
Propósito y contextualización del tema.....	19
Justificación	21
Objetivos.....	22
<i>Objetivo general</i>	22
<i>Objetivos específicos</i>	22
Hipótesis	22
Variables de investigación.....	23

	11
<i>Variable independiente</i>	23
<i>Variable dependiente</i>	23
Metodología	23
Capítulo II: Marco Teórico	25
Sistema de prevención de intrusos	26
Indicadores de Compromiso	26
Modelos y/o algoritmos de Machine Learning	36
Metodología - Scrum	42
Extensiones de Google Chrome.....	43
Herramientas	44
Capítulo III: Introducción.....	47
Implementación del sistema	47
Diseño y Análisis del Sistema	49
Definición de las tecnologías	51
Análisis de Requisitos	55
Lista de Tareas	57
Desarrollo Metodológico.....	59
<i>Visual Studio Code</i>	59
<i>PythonAnywhere</i>	59
<i>Postman</i>	59
<i>GitHub</i>	59
<i>Mockups</i>	60
Implementación de algoritmos y modelos de Machine Learning para sitios web	60

<i>Phishing</i>	60
Sprint 1: Selección del mejor modelo y/o algoritmo de Machine Learning	60
Sprint 02: Creación del dataset	67
Sprint 03: Creación de la API	86
Sprint 04: Desarrollo de la extensión de Google Chrome	93
Ejecución	101
Capítulo IV: Validación del Sistema	110
Definición y evaluación de métricas utilizadas	112
<i>Aplicación de las métricas de evaluación</i>	112
Análisis de resultados	126
Conclusiones	130
Recomendaciones	132
Bibliografía	133
Anexos	139

ÍNDICE DE TABLAS

Tabla 1 <i>Indicadores de Compromiso con respecto a una URL</i>	28
Tabla 2 <i>Características de sitios web a partir de la URL</i>	30
Tabla 3 <i>Modelos y/o algoritmos de Machine Learning,</i>	38
Tabla 4 <i>Fórmulas de métricas de evaluación</i>	48
Tabla 5 <i>Matriz de confusión para Preempt Phisher</i>	49
Tabla 6 <i>Rol de Scrum designados</i>	55
Tabla 7 <i>Historias de usuario</i>	56
Tabla 8 <i>Product Backlog del proyecto</i>	57
Tabla 9 <i>Lista de Tareas</i>	57
Tabla 10 <i>Product Backlog de la Lista de Tareas</i>	58
Tabla 11 <i>Tarea para la selección del Mejor Modelo/Algoritmo y Uso de Indicadores de Compromiso para la Prevención de Phishing en Sitios Web.</i>	61
Tabla 12 <i>Sprint Backlog 01</i>	62
Tabla 13 <i>Resultados de pruebas modelos y/o algoritmos de Machine Learning Implementados.</i>	66
Tabla 14 <i>Tarea para la creación de un dataset</i>	67
Tabla 15 <i>Sprint Backlog 02</i>	68
Tabla 16 <i>Resultados pruebas modelos y/o algoritmos de Machine Learning implementados en diferentes escenarios</i>	77
Tabla 17 <i>Ganador de cada escenario</i>	83
Tabla 18 <i>Tarea para la creación de la API</i>	86
Tabla 19 <i>Sprint Backlog 03</i>	87
Tabla 20 <i>Historia de usuario para el desarrollo de la extensión de Google Chrome</i>	94

Tabla 21 <i>Sprint Backlog 04</i>	95
Tabla 22 <i>Categorización de Sitios Web</i>	112
Tabla 23 <i>Resultados pruebas de Preempt Phishing junto con ZPhisher con modelo entrenado de Machine Learning</i>	114
Tabla 24 <i>Matriz de Confusión del modelo ML usando Zphisher</i>	118
Tabla 25 <i>Métricas de evaluación calculadas</i>	119
Tabla 26 <i>Resultados pruebas de Preempt Phishing junto con MaxPhisher con modelo entrenado de Machine Learning</i>	119
Tabla 27 <i>Matriz de Confusión del modelo ML usando MaxPhiser</i>	124
Tabla 28 <i>Métricas de evaluación calculadas</i>	125
Tabla 29 <i>Comparación de modelo con respecto a Zphisher y MaxPhisher</i>	127

ÍNDICE DE FIGURAS

Figura 1 <i>Diagrama de la arquitectura lógica del sistema</i>	50
Figura 2 <i>Diagrama de la arquitectura lógica del sistema con las tecnologías a usar</i>	52
Figura 3 <i>Diagrama de arquitectura física del sistema.</i>	54
Figura 4 <i>Implementación de modelos y/o algoritmos de Machine Learning</i>	65
Figura 5 <i>Pruebas de características con diferentes escenarios</i>	76
Figura 6 <i>Extracción de Características del sitio web Login Facebook</i>	84
Figura 7 <i>Extracción de Características del sitio web</i>	85
Figura 8 <i>Modelo entrenado y guardado</i>	90
Figura 9 <i>API alojado al Servidor</i>	91
Figura 10 <i>Predicción de sitios web utilizando la API desarrollada</i>	91
Figura 11 <i>Extensión de Google Chrome desarrollada.</i>	99
Figura 12 <i>Notificación enviada al momento de la detección de phishing en el sitio web...</i>	100
Figura 13 <i>Bloqueo de pantalla al sitio web detectado como malicioso.</i>	100
Figura 14 <i>Modelos/Algoritmos de Machine Learning</i>	101
Figura 15 <i>Algoritmo de 40 características para generar Dataset</i>	102
Figura 16 <i>Dataset Creado</i>	103
Figura 17 <i>Pruebas con Respectivos Modelos/Algoritmos de Machine Learning</i>	104
Figura 18 <i>Servidor PythonAnywhere</i>	104
Figura 19 <i>Utilización Postman para verificación respectiva de URL e identificar si tiene Phishing o no</i>	105
Figura 20 <i>Frontend conexión con el Backend</i>	106
Figura 21 <i>Interfaz Gráfica de la Extensión en Google Chrome</i>	107

Figura 22 <i>Ataques disponibles MaxPhisher</i>	110
Figura 23 <i>Ataques disponibles Zphisher</i>	110
Figura 24 <i>Proceso de pruebas</i>	111
Figura 25 <i>Curva de Aprendizaje</i>	125

Resumen

Con el rápido crecimiento tecnológico, el internet se ha convertido en un espacio donde se almacena y recopila grandes volúmenes de información, lastimosamente la plataforma se ha convertido en el principal objetivo de ciberataques. Su infraestructura no es controlada además posee poca seguridad que presenta un conjunto de vulnerabilidades a usuarios, sistemas y amenazas que pueden generar daños financieros, robos de identidad, bancarios o personales. Los Phishers para acceder a datos privados de usuarios, empresas y/u organizaciones utilizan Phishing, que es uno de los ataques más comunes y populares entre todos, el atacante usa envió de correos para ganar premios, envió de mensajes desde cuentas falsas en redes sociales, piratear la contraseña, enviar correos electrónicos a sus víctimas lo cual hace es revelar la información para la ganancia financiera. Los algoritmos de Machine Learning para un Sistema de Prevención de Intrusos (IPS) para la prevención de sitios web con phishing siguen la metodología ágil Scrum. Obtención de características de la URL enfocada principalmente en la obtención de Indicadores de Compromiso para el respectivo entrenamiento del modelo. La aplicación fue probada y validada, en un ambiente de entrenamiento como en un ambiente real/simulado, con ayuda de los simuladores de phishing Zphisher y MaxPhiser, además de su respectiva comparativa con respecto a cada una de las herramientas finalmente se obtuvo resultados aceptables en el rango establecido en artículos en la utilización de Indicadores de Compromiso.

Palabras clave: Machine Learning, sistema de prevención de intrusos, phishing, sitios web, indicadores de compromiso

Abstract

With the rapid technological growth, the Internet has become a space where large volumes of information are stored and collected, unfortunately the platform has become the main target of cyber-attacks. Its infrastructure is not controlled and has little security that presents a set of vulnerabilities to users, systems and threats that can generate financial damage, identity theft, banking or personal. Phishers to access private data of users, companies and / or organizations use Phishing, which is one of the most common and popular attacks among all, the attacker uses sent emails to win prizes, sent messages from fake accounts on social networks, hacking the password, send emails to their victims which does is to reveal the information for financial gain. Machine Learning algorithms for an Intrusion Prevention System (IPS) for the prevention of phishing websites follow the agile Scrum methodology. URL feature elicitation focused on obtaining Indicators of Engagement for the respective training of the model. The application was evaluated and validated, in a training environment as well as in a real/simulated environment, with the help of the phishing simulators Zphisher and MaxPhiser, in addition to their respective comparison with respect to each of the tools, finally acceptable results were obtained in the established range.

Key words: Machine Learning, intrusion prevention system, phishing, web sites, engagement indicators

Capítulo I

Introducción

Propósito y contextualización del tema

Con el rápido crecimiento tecnológico, el internet se ha convertido en un espacio donde se recopilan y guardan volúmenes significativos de datos, debido a su constante evolución. Esto ha provocado que este tipo de plataforma se convierta en el principal objetivo de ciberataques (Sahingoz et al., 2019).

Además de ser una infraestructura no controlada y con poca seguridad que presenta un conjunto de vulnerabilidades a usuarios, sistemas, amenazas que pueden generar daños financieros, robos de identidad, incluso pérdida de usuarios como es en el caso de comercio electrónico (Mohammad et al., 2014; Sahingoz et al., 2019).

Las organizaciones se enfrentan a una ola acelerada de ataques sofisticados por parte de ciberdelincuentes que interrumpen los servicios, extraen datos confidenciales o abusan de máquinas y redes de las víctimas para realizar actividades maliciosas. Para frenar este tipo de retos de seguridad y eliminar los puntos ciegos de la seguridad de sistemas y redes, servicios locales y entornos en la nube se aplican defensas con plataformas de inteligencia de amenazas. La inteligencia sobre ciberamenazas aumenta la visibilidad de las amenazas cibernéticas y las violaciones de las políticas ayuda a reducir el tiempo de detección de amenazas conocidas y desconocidas (Preuveneers et al., 2021).

Algunas de estas amenazas están presentes en Internet, a las cuales se les conocen como amenazas cibernéticas estas pueden ser: Phishing, Malware, Botones, Ransomware, Cryptojacking, Bruce Force Attack. Una de las grandes amenazas, existentes hoy en día, es el Phishing, que están presentes en sitios web (Anupam y Kar,2021).

El phishing es una forma de ciberataque que afecta negativamente a las personas, ya que el usuario es redirigido a sitios web falsos y engañado para que revele su información personal y confidencial, como contraseñas de cuentas, datos bancarios, datos de la tarjeta de débito posiblemente serán usados con fines ilegales (Kumar, K. y Kumar, P. 2020).

Según el informe Cost of a Data Breach Report de IBM 2022 reveló que los ataques de Phishing son la segunda causa más común de vulneración, con un 16%, el vector de ataque más costoso en 2022, en promedio, fue el phishing con 4,91 millones de dólares. Las vulneraciones causadas por el mismo tuvieron el tercer tiempo medio más alto hasta identificar y contener la vulneración con 295 días (CDB, 2022).

Esto demuestra el impacto al recibir un ataque phishing el cual puede llegar a causar una gran pérdida económica individual o dentro de una empresa. En este contexto, la seguridad informática es un factor clave para resguardar la información tanto privada como financiera de usuarios y de empresas (López C. 2016).

En la actualidad, se han realizado estudios e investigaciones sobre cómo detectar, prevenir y/o mitigar los ataques de Phishing, concluyendo que la solución más frecuente entre estas es el uso de diferentes modelos y/o técnicas de Machine y/o Deep Learning o incluso combinaciones entre estas (Sameen et al., 2020).

Sin embargo, la ventaja de compartir los modelos ML frente a los IoC es que ofrecen métodos más sofisticados y eficaces para identificar los ataques se los detecta mediante patrones (características) que permitan distinguir sitios web con Phishing de los que son legítimos, ya que los IoC suelen caracterizar un evento de amenaza como una simple lista de atributos etiquetados y anotados (dirección IP atacante) aunque el valor de los mismos se pueden deteriorar con el tiempo (Preuveneers et al., 2021).

El propósito del presente proyecto es desarrollar un Sistema de Prevención de Intrusos (IPS) enfocado a la prevención de sitios web con Phishing usando Indicadores de Compromiso (IOC), como una extensión de Google Chrome, para la cual se utilizará

modelos y/o algoritmos de Machine Learning (ML), con el objetivo de automatizar el proceso de prevenir sitios web con Phishing, así como también brindar una precisión alta.

Finalmente, la extensión podría dar una solución contra el problema de robo de información que se presentan actualmente en sitios web.

Justificación

En los recientes años, el Internet se ha convertido en una herramienta necesaria e importante en la vida cotidiana, los usuarios de Internet pueden tener una seguridad deficiente para diferentes tipos de amenazas web, lo que provoca a lo largo pérdidas financieras o clientes que carecen de confianza en el comercio y la banca en línea (Karabatak y Mustafa, 2019).

Estas amenazas son aprovechadas por personas mal intencionadas (hackers) que además de generar pérdidas de información producen pérdidas económicas, direccionadas a individuos y/o empresas (Ndichu et al., 2018; Sönmez et al., 2018). Los hackers para acceder a los datos privados de usuarios, empresas, y/u organizaciones utilizan Phishing, que es uno de los ataques más comunes y populares entre todos, el atacante cebo a los usuarios mediante el envío de correos como ganar premios, enviar mensajes desde cuentas falsas en redes sociales, piratear la contraseña, enviar correos electrónicos a las víctimas. Lo cual hace revelar información para la ganancia financiera (Aguilar, 2017; Babar y Ghani, 2021).

Por lo tanto, contar con un Sistema de Prevención de Intrusos (IPS) que detecte el Phishing en sitios web resulta de gran importancia para hacer frente a las amenazas cibernéticas y proteger a los usuarios de Internet. En este proyecto se propone desarrollar dicho IPS, el cual se implementará como una extensión para el navegador Google Chrome. Se ha elegido esta opción debido a su facilidad de instalación, su disponibilidad para cualquier usuario y su integración con el navegador.

Adicionalmente, se busca utilizar modelos y algoritmos de Aprendizaje Automático (Machine Learning) junto con los Indicadores de Compromiso (IOC) para implementar un

sistema de predicción y prevención más eficiente de sitios web de Phishing. Este sistema se basará en la extracción de características específicas de una URL, las cuales serán identificadas a través del análisis exhaustivo de la literatura científica relacionada con este tema. Se dará especial énfasis a aquellas características más relevantes que puedan mejorar las predicciones del IPS y, prevenir estos ataques de Phishing de manera más efectiva.

Objetivos

Objetivo general

Desarrollar un sistema de prevención de intrusos en sitios web, usando indicadores de compromiso aplicando Machine Learning para mejorar la seguridad en la red al proteger información sensible: Caso Práctico Phishing Google Chrome.

Objetivos específicos

- Analizar el estado del arte sobre los indicadores de compromiso y cómo estos pueden ayudar para la prevención de intrusos en páginas o sitios web, apoyado por phishing en motores de búsqueda - Google Chrome.
- Desarrollar una extensión de Google Chrome utilizando técnicas de Machine Learning para mejorar la prevención y gestión de seguridad en sitios web
- Comprobar los resultados obtenidos, analizar y ajustar los errores encontrados en los indicadores de compromiso del sistema de prevención de intrusos.

Hipótesis

El desarrollo de un sistema de prevención de Intrusos en sitios web, usando indicadores de compromiso aplicando técnicas de Machine Learning permite reducir la frecuencia de los ataques a sitios web.

Variables de investigación

Variable independiente

Sistema de prevención de intrusos

Variable dependiente

Indicadores de compromiso aplicando Machine Learning

Metodología

El objetivo principal de este proyecto es crear un sistema de prevención de intrusos que pueda identificar sitios con Phishing, de manera que se puedan alcanzar los objetivos planteados, en esta se utiliza tres fases:

En la Fase I del presente trabajo, se examinará la literatura científica relacionada con el tema propuesto de investigación que abarca indicadores de compromiso con el fin de desarrollar el marco teórico. Para lograr esto, se usó métodos teóricos que son comunes en la investigación científica, como el método análisis-síntesis.

También se analizará las características de los sitios web, enfocándonos en los recursos de verificación que permiten determinar si un sitio web posee o no Phishing. Esto se basará en el análisis de las URL orientadas principalmente en Indicadores de Compromiso (IOC).

Para llevar a cabo esta investigación, se usó la base de datos bibliográfica SCOPUS, Data Science y el repositorio MISP, KAGGLE

En la Fase II se abarca sobre la implementación del sistema de prevención de intrusos junto con la extensión de Google Chrome, utilizando estándares, técnicas y metodologías respectivas para el desarrollo de software. Para comenzar, se escogen data sets que incluyen todas las características necesarias. Luego se generan escenarios, variando la cantidad de características, sitios web legítimos y con phishing, y combinación de escenarios.

Se procederá a implementar los modelos de Machine Learning previamente seleccionados de la fase anterior, para luego realizar pruebas en cada uno de los escenarios junto con cada modelo implementado. Se realizarán los ajustes necesarios de acuerdo con los resultados obtenidos. Además, se creará un conjunto de datos propio utilizando un dataset existente que contenga URL legítimas y con phishing, utilizando el código de extracción de características desarrollado previamente.

Esto ayudará a entrenar el modelo de predicción con datos actuales y llevar a cabo pruebas en el modelo utilizando un simulador de sitios web con phishing. Finalmente, se implementará y desplegará una extensión de Google Chrome capaz de realizar solicitudes al modelo de predicción desarrollado y cargado en el servidor.

En la Fase final, se busca verificar la eficacia del IPS a través de métodos experimentales y empíricos. El objetivo es confirmar los resultados una vez que el sistema de prevención de intrusos se haya implementado y lanzado al público.

Capítulo II

Marco Teórico

En este capítulo se lleva a cabo una investigación teórica sobre los sistemas de prevención de intrusiones (IPS), Indicadores de Compromiso (IOC), los componentes de verificación (características), los modelos y algoritmos de Aprendizaje Automático, la Metodología y las métricas de precisión. Estos elementos son fundamentales para desarrollar un sistema de prevención de phishing que permita determinar si un sitio web contiene o no contenido de phishing.

Para abordar este tema, se llevó a cabo una breve exploración de la literatura utilizando la base de datos bibliográfica de resúmenes y citas de artículos de revistas conocida como SCOPUS. Inicialmente, se creó una cadena de búsqueda que incluía términos relacionados con el objeto de estudio propuesto. Después de ingresar la cadena de búsqueda en SCOPUS, se llevó a cabo el proceso de revisión y selección de artículos pertinentes para la investigación, basándonos principalmente en tres criterios.

En primer lugar, los artículos debían estar directamente relacionados con el tema de investigación. En segundo lugar, se consideró que los artículos debían tener un número de citas igual o superior a 7 en promedio. Por último, se limitó la selección a artículos publicados en el período comprendido entre 2017 y 2022.

Siguiendo estos criterios, se identificaron 11 artículos relevantes, cuya información se detalla en el Anexo 1, se utilizó la plataforma de Data Science para buscar artículos relevantes sobre las características más frecuentemente empleadas en la detección de phishing en sitios web. Además, se exploró el repositorio KAGGLE en busca de conjuntos de datos relacionados con el tema de investigación. Al ingresar las palabras clave “features”, “phishing” y “detection”, se obtuvieron un total de 11 conjuntos de datos, como se muestra en un estudio previo (Castillo Veloz y Chuquitarco Veloz, 2023).

Para finalizar el capítulo, se incluye información sobre las extensiones de Google Chrome, con el objetivo de abordar completamente el tema de este trabajo.

Sistema de prevención de intrusos

Un sistema de detección de intrusos (IDS, por sus siglas en inglés) es una herramienta crucial para proteger los sistemas y redes de ataques maliciosos. Se encarga de monitorear y analizar el tráfico de red en busca de actividades sospechosas o patrones anómalos que puedan indicar una intrusión. Uno de los enfoques comunes en la detección de intrusos es el uso de algoritmos de aprendizaje automático, como las redes neuronales artificiales (Smith et al., 2019). La detección de intrusos es parte fundamental de la seguridad de la información en entornos computacionales. Un enfoque comúnmente utilizado en la detección de intrusos es el uso de sistemas basados en firmas, los cuales comparan el tráfico de red con una base de datos de patrones conocidos de ataques (García et al., 2021). En la actualidad, el phishing se ha convertido en el ataque preferido por los hackers debido a su facilidad de ejecución y alta efectividad. Según un estudio de ciberataque hecho en 2017, por la empresa INSIBE Instituto de Seguridad de España, el ataque más común es el Phishing (Macancela et al., 2019). Los piratas informáticos utilizan diversas técnicas para engañar a sus víctimas, siendo una de las más comunes la redirección a sitios web falsos. Una vez en estos sitios, las personas son inducidas a ingresar sus datos personales y confidenciales, lo que permite a los atacantes obtener información valiosa para llevar a cabo robos de dinero y suplantaciones de identidad (Aguilar, 2017).

Indicadores de Compromiso

Los indicadores de compromiso en software son métricas o señales que indican la presencia de una brecha de seguridad o una posible intrusión en un sistema o aplicación (Smith et al., 2022).

Estos indicadores son fundamentales para detectar y responder a actividades maliciosas, y pueden ayudar a las organizaciones a proteger sus activos digitales. Algunas áreas clave relacionadas con los indicadores de compromiso en software:

- Registro y monitoreo de eventos: El análisis de registros y eventos es una técnica esencial para identificar indicadores de compromiso en software. Los registros de eventos, como los registros de auditoría y los registros de seguridad, pueden proporcionar información valiosa sobre actividades sospechosas, intentos de acceso no autorizado o comportamientos anómalos. El monitoreo en tiempo real de estos eventos permite una respuesta rápida ante posibles compromisos (García, L.et. 2021)
- Análisis de tráfico de red: El análisis del tráfico de red puede revelar patrones de comunicación sospechosos o anómalos. Los indicadores de compromiso en esta área incluyen el descubrimiento de comunicaciones no autorizadas, el análisis de paquetes de red maliciosos o el seguimiento de patrones de tráfico inusuales que podrían indicar actividades de intrusión (Chen et al., 2020).
- Detección de malware: Los indicadores de compromiso son una buena estrategia para identificar el Malware porque están compuestos de firmas de virus, los cuales son piezas de código del propio virus para crear los IOC, seleccionar dominios, direcciones web maliciosas y direcciones IP con el objetivo fin de prevenir e identificar amenazas (Larreategui. 2021).
- Análisis de vulnerabilidades: La identificación y el seguimiento de vulnerabilidades conocidas en el software son indicadores importantes de compromiso. Estos indicadores pueden basarse en bases de datos de vulnerabilidades, análisis de parches o información de proveedores de seguridad (Thompson et al., 2019).

A continuación, se presenta la tabla 1 que resume algunos de los indicadores de compromiso comúnmente observados en las URLs desglosando como puntos principales la descripción y la iniciativa la cual abarca a los autores respectivos que mencionan cada una

de las características utilizadas ver Anexos 1. Estos indicadores ayudan a evaluar la confiabilidad de un sitio web antes de interactuar con él.

Tabla 1

Indicadores de Compromiso con respecto a una URL

Ord.	Recurso de Comprobación	Descripción	Iniciativa
1	MD5	Es un algoritmo de hash criptográfico ampliamente utilizado en la detección de phishing. Proporciona una forma de verificar la integridad de un archivo o un conjunto de datos al generar un valor hash único de 128 bits.	S07
2	SHA1	En el contexto de la detección de phishing, el SHA-1 puede ser utilizado para calcular el hash de una URL o un sitio web sospechoso y compararlo con una base de datos de hash conocidos de sitios de phishing.	S08
3	YARA	Utiliza reglas y patrones definidos por el usuario para buscar coincidencias en archivos y procesos. Estas reglas se basan en características específicas del código malicioso, como cadenas de texto, firmas digitales y comportamientos conocidos. Al aplicar estas reglas, YARA puede identificar y marcar archivos sospechosos o maliciosos.	S09
4	SHA256	Permite comparar el hash de un sitio web sospechoso con el hash de un sitio web legítimo conocido. Si los hashes coinciden, se considera	S10

Ord.	Recurso de Comprobación	Descripción	Iniciativa
		que el sitio web es legítimo.	
5	IPSrc	La dirección IP de origen puede ser utilizada como uno de los muchos atributos o características para identificar y analizar posibles actividades de phishing. Por ejemplo, se puede utilizar para rastrear la ubicación geográfica del origen de una solicitud de phishing o para detectar patrones de comportamiento sospechosos.	S11
6	Domain	Consiste en identificar los dominios maliciosos utilizados en ataques de phishing y utilizarlos como indicadores para bloquear o advertir a los usuarios sobre posibles intentos de phishing.	S12
7	Hostname	Se analiza el hostname del sitio web en cuestión para identificar patrones o características que sean indicativos de actividades fraudulentas. Esto puede incluir la presencia de palabras clave relacionadas con phishing, la utilización de dominios engañosos o similares a sitios legítimos, o la inclusión de números o caracteres especiales que buscan imitar una URL auténtica.	S13

Nota. Se presenta los indicadores de compromiso con respecto a la URL utilizadas.

Características para detección de intrusos - Phishing

Tras realizar el estudio de la literatura, se recopiló un total de 62 características, que se detallan en el Anexo 3. Sin embargo, se encontró muchas características demasiado específicas, por lo que se optó por agruparlas de manera global, como se puede apreciar en

el Anexo 2: Datasets seleccionados en la revisión de la literatura implementada. Finalmente, se seleccionaron un total de 30 características, basándose en su frecuencia de aparición, la cual debía ser igual o superior al valor medio de la frecuencia total.

Estas características seleccionadas se presentan en orden descendente de frecuencia en la tabla 2. Además, de algunas características que se mencionaron en la tabla 1, se presenta la descripción de los recursos de comprobación, también la iniciativa que hace énfasis a los autores que destacan cada uno de los puntos mencionados ver Anexos 1

Tabla 2

Características de sitios web a partir de la URL.

Ord.	Recursos de Comprobación	Descripción	Iniciativa
1	Presencia de dirección IP	La probabilidad de que un sitio web sea víctima de phishing aumenta cuando se utiliza una dirección IP en lugar del nombre de dominio en una URL.	S01, S02, S03, S04, S05, S06
2	Longitud de la URL	Los sitios web de phishing intentan ocultar el nombre de dominio al utilizar URL largas. En general, las URL de los sitios web legítimos suelen tener una longitud inferior a 54 caracteres. Si una URL excede los 75 caracteres, existe una alta probabilidad de que se considere phishing, y si iguala o supera los 75 caracteres, se considera completamente phishing.	S01, S02, S03, S04, S05, S06
3	Presencia del símbolo @	La presencia del símbolo "@" en una URL provoca que el navegador web ignore todo lo que se encuentra antes de dicho símbolo, lo	S01, S02, S03, S04, S05, S06

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		cual aumenta la probabilidad de que se redirija a sitios web con phishing.	
4	Estado SSL	El enfoque consiste en determinar si un sitio web cuenta con un certificado SSL, ya que los sitios web de phishing no encriptan los datos enviados y, por lo tanto, no poseen dicho certificado. Para identificar esto, se verifica si la URL comienza con "HTTPS" y si sus proveedores son confiables.	S01, S02, S03, S04, S05, S06
5	Edad del Dominio	La valoración de un sitio web puede basarse en la duración de su dominio. Si el tiempo transcurrido desde su creación es inferior a 6 meses, existe una alta probabilidad de que sea un sitio web con phishing.	S01, S02, S03, S04, S05, S06
6	Redirecciones de doble barra	Si la posición de los símbolos "/" en una URL es mayor a 7, se considera un indicio de que se trata de un sitio web de phishing. Esto se debe a que, en los sitios legítimos, generalmente se utiliza la redirección de doble barra solo una vez.	S01, S02, S03, S04, S05, S06
7	URL de anclaje	Este método implica contar la cantidad de veces que las etiquetas <a> con enlaces dentro del sitio web redirige a un dominio distinto al propio. Si dicha cantidad supera el valor del 31%, se considera que el sitio web tiene características sospechosas de phishing.	S01, S02, S03, S04, S05, S06

Ord.	Recursos de Comprobación	Descripción	Iniciativa
8	Prefijo/ Sufijo	Los sitios web de phishing suelen utilizar el símbolo "-" para añadir prefijos y sufijos a las URL, mientras que los sitios web legítimos generalmente no utilizan este símbolo.	S01, S02, S03, S04, S05, S06
9	Enlaces en etiquetas	Se realiza un análisis exhaustivo de todas las etiquetas presentes en el sitio web y hacia dónde se dirigen. Si se encuentra que alguna de las etiquetas redirige a un sitio web presente en la lista negra, se considera que el sitio web está involucrado en actividades de phishing.	S01, S02, S03, S04, S05, S06
10	Deshabilitar clic derecho	En la actualidad, es común que los sitios web legítimos desactiven la opción de clic derecho para evitar que los usuarios realicen modificaciones en el código fuente del sitio.	S01, S02, S03, S04, S05, S06
11	Uso de ventana emergente	Las ventanas emergentes se muestran en la pantalla con un menú y desaparecen al hacer clic en ellas. Una señal de que un sitio web puede estar involucrado en phishing es cuando solicita información al usuario a través de ventanas emergentes.	S01, S02, S03, S04, S05, S06
12	Favicon	El favicon es un icono utilizado para identificar un sitio web de manera rápida. Si el favicon de un sitio web es diferente al dominio que se muestra en la URL, existe una alta probabilidad de que dicho sitio web esté relacionado con	S01, S02, S03, S04, S05, S06

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		actividades de phishing.	
13	URL anormal	Se realiza una verificación para determinar si la URL contiene el nombre del host y si este coincide con el dominio mostrado en la URL. En caso de que la URL no posea estas dos características, se considera que el sitio web está involucrado en actividades de phishing.	S01, S02, S03, S04, S05, S06
14	IFrame	Las etiquetas IFrame se utilizan para redirigir a los usuarios a otro sitio web dentro de un mismo sitio web. Estas etiquetas pueden ser utilizadas de manera engañosa para confundir a los usuarios.	S01, S02, S03, S04, S05, S06
15	Registro DNS	El registro DNS de un sitio web contiene información crucial, por lo tanto, los sitios web de phishing tienden a ocultar este registro para evitar ser detectados.	S01, S02, S03, S04, S05, S06
16	Índice de Google	Debido a su corta vida útil, los sitios web de phishing no son indexados por Google.	S01, S02, S03, S04, S05, S06
17	Puerto Utilizado	Los puertos considerados confiables son el 8080 y el 443. Si un sitio web utiliza un puerto diferente, existe una alta probabilidad de que se trate de un sitio web de phishing.	S01, S02, S03, S04, S05, S06
18	Request URL	Se lleva a cabo una inspección para determinar si los objetos externos contenidos en una página	S01, S02, S03, S04,

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		web se cargan desde un dominio diferente. Si la dirección URL se encuentra fuera del dominio actual, se considera que el sitio web está involucrado en actividades de phishing.	S05, S06
19	SFH (Controlador de Formulario de Servidor)	El enfoque se centra en la gestión de formularios y se verifica si al completar el formulario, el botón de envío (submit) devuelve un mensaje vacío o no proporciona ninguna puesta. Esto se realiza para identificar posibles etiquetas asociadas a sitios web de phishing.	S01, S02, S03, S04, S05, S06
20	Recuento de redirección del sitio web	Este enfoque se basa en verificar la cantidad de veces que las fuentes (enlaces, imágenes, etc.) redirigen a una única dirección web o a un sitio web con un dominio diferente al que se muestra en la barra de búsqueda.	S01, S02, S03, S04, S05, S06
21	MouseOver	Esta funcionalidad solía mostrar información del sistema en la parte inferior de la pantalla. Sin embargo, en la mayoría de los sitios web legítimos actualmente ya no se utiliza, por lo que si un sitio web la emplea, se considera sospechoso.	S01, S02, S03, S04, S05, S06
22	Trafico Web	La evaluación de un sitio web puede basarse en la cantidad de visitas que recibe diaria, semanal o mensualmente. Por lo general, cuanto mayor sea este valor, se considera que el sitio web es	S01, S02, S03, S04, S05, S06

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		más confiable.	
23	Servicio de Acortamiento	Un servicio de acortamiento de URL es una técnica utilizada para abreviar una URL y redirigir a la misma página que la dirección original. La mayoría de los sitios web de phishing hacen uso de este tipo de servicios.	S01, S02, S03, S04, S05, S06
24	Duración de Registro de Dominio	La información requerida se extrae del registro Whois, donde se analiza el período de tiempo durante el cual el dominio de un sitio web ha sido registrado. Si el número de años registrado es igual o menor a 1 año, se considera que el sitio web está relacionado con actividades de phishing.	S01, S02, S03, S04, S05, S06
25	Token HTTPS	Se hace referencia al uso combinado del protocolo TLS/SSL en conjunto con el HTTP seguro.	S01, S02, S03, S04, S05, S06
26	Envío de información al correo electrónico	Se realiza una verificación para determinar si el sitio web utiliza algún tipo de servicio "mail () to" internamente, lo cual podría indicar que se trata de un sitio web de phishing.	S01, S02, S03, S04, S05, S06
27	Rango de Página	El cálculo del rango de un sitio web se basa en contar los enlaces salientes y entrantes presentes en él, lo cual representa su nivel de importancia. Si este valor es menor a 0.2, existe una alta sospecha de que el sitio web esté	S01, S02, S03, S04, S05, S06

Ord.	Recursos de Comprobación	Descripción	Iniciativa
		involucrado en actividades de phishing.	
28	Informe Estadístico	Los informes estadísticos proporcionan datos sobre sitios web legítimos y aquellos relacionados con phishing, además de otros datos estadísticos relevantes.	S01, S02, S03, S04, S05, S06
29	Presencia de Subdominio	Un indicio de que un sitio web es víctima de phishing es si contiene más de 2 subdominios en su URL. Para identificar esto, es necesario observar la cantidad de puntos presentes en el dominio, ya que, si este número es mayor a 2, se considera que el sitio web es víctima de phishing.	S01, S02, S03, S04, S05, S06
30	Enlaces que apuntan a la página	La autenticidad de un sitio web se puede determinar evaluando la cantidad de enlaces que se dirigen hacia ese sitio. Para ser considerado legítimo, el sitio web debe contar con al menos 2 enlaces que lo respalden.	S01, S02, S03, S04, S05, S06

Nota. Se presenta las características de sitios web con respecto a la URL utilizadas.

Modelos y/o algoritmos de Machine Learning

El Machine Learning, se refiere a un campo de investigación cuyo propósito es capacitar a las computadoras para que puedan adquirir conocimiento a partir de conjuntos de datos y tomar decisiones o realizar predicciones sin necesidad de programación explícita (Mahesh, 2019; Ray, 2019; Song et al., 2017).

El Machine Learning encuentra aplicaciones en diversos campos científicos como la robótica, los videojuegos, el reconocimiento de patrones, la minería de datos, el procesamiento de lenguaje natural, la medicina, la seguridad informática, el reconocimiento, entre otros (Akinsola, 2017). Los algoritmos de Machine Learning se refieren a conjuntos de instrucciones que permiten analizar conjuntos de datos y crear modelos capaces de predecir o clasificar información (Hao y Ho, 2019).

Los algoritmos y/o modelos de Machine Learning pueden adoptar diferentes enfoques de aprendizaje, como el supervisado y el no supervisado. El aprendizaje supervisado enseña al algoritmo como realizar su trabajo (Rojas, E. 2018).

Con un conjunto de datos clasificados bajo cierta apreciación o idea para encontrar patrones que se puedan aplicar en un análisis (Mueller y Massaron 2016).

Algunos algoritmos comunes utilizados en el aprendizaje supervisado incluyen árboles de decisión, Support Vector Machines (SVM), Naive Bayes, entre otros. En contraste, el aprendizaje no supervisado se enfoca en descubrir patrones en conjuntos de datos que no están etiquetados y posteriormente clasificarlos (Rajoub, 2020; Ray, 2019).

Existen dos tipos de modelos y/o algoritmos de Machine Learning:

regresión/clasificación y agrupación/reducción. Los modelos de regresión/clasificación se basan en el aprendizaje supervisado, utilizando datos de entrada y salida para realizar el entrenamiento y las pruebas, lo que resulta en una mayor eficiencia en los procedimientos (Yuan et al., 2012).

Por otro lado, los modelos de agrupación/reducción pertenecen al aprendizaje no supervisado y son ampliamente utilizados para el análisis de datos (Gates y Ahn, 2017). En este tipo de modelos, los datos de entrada no tienen una categoría definida, y el algoritmo se encarga de buscar patrones y agruparlos en diferentes categorías.

Según la revisión de la literatura presentada en Anexo 1 se han identificado los algoritmos y/o modelos más utilizados para la detección de Phishing. Se analizó tanto la precisión máxima alcanzada como la incidencia de estos modelos en la detección.

A continuación, en la tabla 3, se detallan los algoritmos y/o modelos seleccionados encontrados con su respectiva descripción.

Tabla 3

Modelos y/o algoritmos de Machine Learning,

Ord.	Modelo y/o algoritmo	Descripción
1	Árbol de Decisión	Modelos de tipo jerárquico que toman decisiones basadas en preguntas sobre características de los datos. El árbol de decisión es un algoritmo de aprendizaje supervisado ampliamente utilizado para realizar tareas de clasificación y regresión. Se caracteriza por su estructura jerárquica, donde los datos se clasifican desde la raíz (nodo raíz) hasta los nodos hoja. Cada nodo del árbol representa una condición, mientras que los nodos hoja representan las respuestas correspondientes a esas condiciones. El proceso de clasificación/regresión comienza en el nodo raíz y se propaga a través del árbol hasta llegar a un nodo hoja, donde se realiza la clasificación o regresión final (Alzubi et al., 2018; Ray, 2019). Según la revisión de la literatura realizada, se ha observado que los árboles de decisión pueden alcanzar una alta precisión, como un máximo del 96,60%, en la detección de phishing.
2	Random Forest:	Un algoritmo que combina múltiples árboles de decisión para realizar predicciones más precisas. Random Forest es uno de los algoritmos de aprendizaje supervisado más populares debido a su sencillez y precisión. Se utiliza para la clasificación y regresión de datos y se basa en la construcción de múltiples árboles de decisión para obtener una salida que combina las

Ord.	Modelo y/o algoritmo	Descripción
		<p>predicciones de cada árbol. Random Forest consta de dos etapas: la primera etapa implica la creación de los bosques aleatorios, y la segunda etapa implica realizar predicciones utilizando el clasificador de bosques aleatorios (Alzubi et al., 2018). Según la revisión de la literatura, se ha observado que Random Forest puede alcanzar una alta precisión, con un máximo del 99,33%, en la detección de phishing.</p>
3	Redes Neuronales	<p>Modelos inspirados en el funcionamiento del cerebro humano, que constan de capas de neuronas interconectadas. Las redes neuronales han ganado popularidad entre los investigadores debido a su comparación con el funcionamiento del cerebro humano, lo que las ha posicionado como un enfoque moderno de aprendizaje (Petersen, 2022). Estas redes tienen como objetivo enseñar a las computadoras a procesar datos de manera similar al cerebro humano, utilizando nodos interconectados que imitan las neuronas. A través del aprendizaje de errores, las redes neuronales mejoran continuamente y se utilizan para abordar problemas complejos con el fin de lograr una mayor precisión (Lauzon, 2012). Según la revisión de la literatura, las redes neuronales han alcanzado una precisión máxima del 97% en la detección de phishing.</p>
4	Support Vector Machines (SVM)	<p>Algoritmos que buscan encontrar el hiperplano óptimo que separa datos en distintas categorías. El algoritmo de Máquinas de Vectores de Soporte (SVM) es un enfoque de aprendizaje supervisado en Machine Learning que puede utilizarse para</p>

Ord.	Modelo y/o algoritmo	Descripción
		clasificación y regresión. Su funcionamiento se basa en la creación de un plano en el espacio de acuerdo con las características de los datos de entrenamiento. Luego, los datos son categorizados utilizando un separador, lo que permite al algoritmo recibir nuevas características y predecir la categoría a la que pertenecen (Ray, 2019). En la detección de phishing, se ha observado que SVM alcanza una precisión máxima del 96,5% según la revisión de la literatura.
5	Mezcla de Gaussianas - GMM	Modelo probabilístico para la distribución de datos en un espacio multidimensional como una combinación ponderada de distribuciones Gaussianas individuales, son utilizados en tareas de agrupamiento y modelado de datos (MacQueen, 1967). Pueden ser aplicados en una variedad de campos, como reconocimiento de patrones, análisis de imágenes, y más (Bishop Y Nasrabadi, 2006).
6	Naive Bayes	Algoritmo de aprendizaje automático basado en el Teorema de Bayes, que asume una independencia condicional “ingenua” entre las características, calcula la probabilidad de que una instancia pertenezca a una determinada clase utilizando la regla de Bayes y toma la probabilidad de cada característica dado un cierto valor de clase (Barber, 2012).
7	Redes Bayesianas	Modelo probabilístico que representa las relaciones de dependencias entre variables utilizando un grafo dirigido acíclico, donde los nodos representan las variables y las aristas

Ord.	Modelo y/o algoritmo	Descripción
		representan las relaciones probabilistas entre ellas (Denis, 2021).
8	Redes Multicapa: MLP	Las redes multicapas estas redes consisten en múltiples capas de unidades llamadas neuronas, organizadas en una estructura en la que las salidas de una capa se convierten en las entradas de la siguiente capa (Heaton, 2018).
9	K - Means	Es un método de agrupamiento (clustering) que se utiliza para dividir un conjunto de datos en grupos o clusters, donde los elementos dentro de cada grupo son similares entre sí y diferentes de los elementos en otros grupos, el objetivo es asignar cada punto al cluster cuyo centroide es el más cercano, y luego ajustar los centroides para minimizar la suma de las distancias al cuadrado entre los puntos y sus centroides correspondiente (Hastie et al., 2009).

Nota. Se presenta los modelos y/o algoritmos de Machine Learning junto con su respectiva descripción.

Basándose en las descripciones detalladas en la tabla 3 de cada algoritmo de Machine Learning que utilizó, se concluye que el algoritmo Random Forest alcanza una precisión máxima del 99.33%. Esto demuestra que posee fundamentos sólidos y lo convierte en una opción destacada en comparación con los otros algoritmos. Al utilizar Random Forest, se podrá identificar eficazmente URLs con contenido de phishing, ya que su elevada precisión garantiza resultados confiables. Además, este algoritmo proporciona un excelente rendimiento en el entrenamiento del modelo, lo que nos permitirá crear un

clasificador altamente preciso y efectivo. Por tanto, se puede afirmar que el uso de Random Forest es altamente recomendado para nuestros propósitos.

Metodología - Scrum

Se usó la metodología ágil, en la cual se realizan iteraciones y pruebas continuas durante todo el ciclo de vida del desarrollo de software (SDLC). Scrum es la metodología ágil comúnmente utilizada para el desarrollo de productos software (Cobb y Charles ,2017), es un marco ágil que proporciona pasos para gestionar y controlar el proceso de desarrollo de software (Schwaber y Mike ,2002).

Se basa en iteraciones que cuenta con un pequeño equipo que trabaja en una tarea asignada. Tiene una duración de 1 a 3 semanas (Rubin y Kenneth S,2012).

Al finalizar el Sprint, se entrega un producto o resultado funcional que pueda ser evaluado por el usuario antes de procesar el siguiente sprint.

El sprint es un contenedor para todos los demás eventos. Cada evento en Scrum es una oportunidad formal para inspeccionar y adaptar los artefactos Scrum. Estos eventos están diseñados específicamente para permitir la transparencia necesaria.

El sprint es un contenedor para todos los demás eventos, los cuales son utilizados para crear regularidad y reducir al mínimo la necesidad de reuniones no definidas. De manera óptima, todos los eventos se llevan a cabo en el mismo momento y lugar para reducir la complejidad (Schwaber, KSutherland, J,2017).

Los eventos más destacados que se presentan en esta metodología son los siguientes:

- **Sprint:** Se refiere a un periodo de tiempo que es generalmente entre 15 a 30 días, durante el cual se desarrolla una funcionalidad o un incremento del

sistema. Durante un sprint, se trabaja para entregar un producto funcional que aporte un valor tangible al sistema.

- **Sprint Planning (Reunión de planificación del sprint):** Se lleva a cabo una reunión formal en la que participan todos los miembros del equipo de desarrollo. El objetivo de esta reunión es organizar las historias de usuario las cuales serán abordadas en el sprint que se va a desarrollar. La duración de esta reunión no debe exceder las dos horas antes de cada inicio de sprint.
- **Daily Scrum (Reuniones Diarias):** Se trata de reuniones cortas que se realizan diariamente, con una duración de 15 minutos como máximo, y en las que participan todos los miembros del equipo de desarrollo. Estas reuniones tienen como objetivo hacer un seguimiento del progreso del sprint, compartiendo actualizaciones y coordinando el trabajo en equipo
- **Sprint Review (Revisión del sprint):** Al finalizar el sprint, se lleva a cabo una reunión de revisión en la que se examina el trabajo realizado por el equipo de desarrollo. El objetivo de esta reunión es revisar todo lo logrado durante el sprint e identificar oportunidades de mejora tanto para el equipo como para el proyecto en general.
- **Sprint Retrospective (Retrospectiva del Sprint):** Se basa en una reunión donde el equipo Scrum planifica formas de aumentar la eficacia del siguiente sprint, basándose en la experiencia adquirida durante el sprint anterior. Durante esta reunión, el equipo discute los problemas encontrados, los logros alcanzados y las metas futuras, aplicando los cambios necesarios para mejorar la efectividad del próximo sprint.

Extensiones de Google Chrome

Las extensiones de Google Chrome son pequeñas aplicaciones desarrolladas para mejorar la funcionalidad y personalización del navegador web Google Chrome. Estas extensiones permiten a los usuarios agregar nuevas características y

funcionalidades a su experiencia de navegación, lo que les permite adaptar el navegador a sus necesidades específicas (Hansen, 2019).

Se ejecutan en el entorno del navegador web y tienen acceso limitado a los recursos del sistema, lo que garantiza la seguridad y privacidad de los usuarios. Además, las extensiones pueden interactuar con las páginas web que los usuarios visitan, agregando elementos adicionales, modificando el contenido existente o mejorando la navegación y la experiencia de usuario de alguna manera (Hansen, 2019).

Algunas extensiones que han sido evaluadas y han demostrado buenos resultados en la detección de phishing son: PIXM Phishing Protection, My Wot, Retruster Phishing Protection, las cuales están disponibles para instalarse en navegadores basados en Chromium. Además, existe la extensión PhishWall que puede ser instalada en el navegador Firefox, entre otras opciones.

El presente estudio propone la creación de una extensión para el navegador Google Chrome con el objetivo de detectar sitios web de phishing. Para llevar a cabo este desarrollo, se requiere tener conocimientos en tecnologías como HTML, CSS, JavaScript y JSON (Mehta, 2016). Las extensiones pueden ser desarrolladas en cualquier sistema operativo, y en comparación con las extensiones de otros navegadores, las extensiones de Google Chrome son más populares y ampliamente utilizadas. Hasta abril de 2023, Google Chrome es el navegador más popular, con un uso del 63.45% (Statcounter Global Stats - Browser, OS, Search Engine Including Mobile Usage Share, s. f.).

Herramientas

Las herramientas que se usarán en el desarrollo del sistema como primordial es Visual Studio Code el cual es un popular y versátil editor de código desarrollado por Microsoft. Proporciona una amplia gama de funciones y extensiones la convierte en una potente herramienta para desarrolladores. Según un estudio realizado por Stack Overflow en 2021, Visual Studio Code se clasificó como el entorno de desarrollo más

popular su interfaz intuitiva, sus sólidas capacidades de depuración y amplia compatibilidad con diferentes lenguajes de programación (Stack Overflow, 2021)

PythonAnywhere es un entorno de desarrollo integrado (IDE) basado en la nube permite a los usuarios escribir, ejecutar y desplegar aplicaciones Python en línea. Ofrece varias características como la edición de código, entornos de ejecución y acceso a base de datos, por lo que es adecuado tanto para principiantes como para desarrolladores experimentados.

JavaScript es un lenguaje de programación muy entendido que se usa principalmente para desarrollar páginas y aplicaciones web interactivas. Como se describe en el libro “JavaScript: The Good Parts” de Douglas Crockford, JavaScript es un lenguaje potente y expresivo que permite a los desarrolladores crear sofisticadas funcionalidades del lado del cliente, lo que convierte en una herramienta esencial para el desarrollo web (Crockford, 2008).

Python es un lenguaje de programación de alto nivel y propósito general conocido por su sencillez y legibilidad. La sintaxis limpia de Python y su amplia biblioteca de estándar lo convierten en un lenguaje versátil utilizado en diversos ámbitos, como el desarrollo web, la informática científica, la inteligencia artificial y el análisis de datos. Como se afirma en el libro “Python Crash Course” de Erick Matthes, la filosofía de diseño de Python hace hincapié en legibilidad y simplicidad del código, por lo que es una excelente opción tanto para principiantes como para programadores experimentados (Matthes, 2019).

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en HTML o XML. Ofrece a los desarrolladores la posibilidad de controlar el aspecto visual de páginas web. En el libro “CSS: The Definitive Guide” de Eric A. Meyer y Estelle Weyl, CSS se describe como una tecnología fundamental para el diseño web, permite a los desarrolladores separar el contenido de su presentación, lo que facilita el mantenimiento y la actualización de la apariencia del sitio web (Meyer y Weyl, 2017).

HTML (Hypertext Markup Language) es un lenguaje de marcado estándar utilizado para crear páginas y aplicaciones web, define la estructura y el contenido de una página web, incluidos títulos, párrafos, imágenes, enlaces y otros elementos. Como se explica en el libro “HTML and CSS: Design and Build Websites” de Jon Duckett, el HTML constituye la columna vertebral de la web y sirve como bloque de construcción para organizar y presentar la información en Internet (Duckett, 2011).

Postman es una popular plataforma de colaboración para el desarrollo de API. Ofrece una interfaz fácil de usar que permite a los desarrolladores diseñar, probar y documentar APIs de manera eficiente. Con funciones como la creación de solicitudes, las pruebas automatizadas y colaboración en equipo. Postman simplifica el proceso de desarrollo e integración de API.

Capítulo III

Introducción

Se detalla todos los procesos llevados a cabo para el desarrollo del sistema propuesto, el cual consiste en un sistema de prevención de phishing utilizando Indicadores de Compromiso mediante el uso de modelos y/o algoritmos de Machine Learning. Este sistema se implementa como una extensión de Google Chrome, con la capacidad de notificar a los usuarios si está visitando un sitio web legítimo o uno con phishing. Además, se describen los pasos desarrollados para crear la API Rest, muestra de una manera visual las interfaces de la extensión, cabe destacar que se conocerá sobre la arquitectura que se utiliza para el debido proceso de desarrollo del sistema propuesto y la forma de la toma de requisitos. Para la creación respectiva de historias de usuario aplicando la metodología propuesta.

Implementación del sistema

Con el objetivo de obtener una comprensión del funcionamiento del sistema de prevención propuesto, se proporcionará una breve explicación de su proceso: en primer lugar, se envía la URL del sitio web a la API, la cual se encargará de extraer las características según lo establecido en la tabla 1. A continuación, se realiza la predicción del sitio web, indicando si contiene phishing o no. Esta información se muestra al usuario a través de la extensión desarrollada para Google Chrome.

Un aspecto para considerar está relacionado con las medidas de evaluación aplicadas a los distintos modelos de Machine Learning que se probaron durante el desarrollo de este proyecto. Esto se llevó a cabo en concordancia con (Robalino, Monleón-Getino y Rodellar,2020) se definieron las siguientes métricas de evaluación:

Accuracy: Hace referencia al porcentaje de elementos correctamente clasificados entre positivos y negativos. Facilidad de cálculo y comprensión para evaluar la efectividad general del algoritmo (Robalino et al., 2020).

Precisión: Mide el porcentaje de verdaderos positivos correctos dividido por el número total de predicciones positivas identificadas.

Recall: Calcula el número de todos los elementos detectados correctamente en proporción a todos los elementos que deben detectarse. Un investigador aumenta sin afectar el valor de la accuracy (Drzewiecki, 2017).

F1: Fusiona las métricas accuracy con recall, presentando diferencias en el rendimiento de un clasificador que no son revelados únicamente con el accuracy. Demuestra que el algoritmo de clasificación predice de mejor manera la clase positiva (Bekkar, Djemaa, y T. Alitouche, 2013)

En la tabla 4, se muestra las fórmulas aplicadas en la evaluación del modelo, mismas que están acordes con (Robalino et al., 2020).

Tabla 4

Fórmulas de métricas de evaluación

MÉTRICA DE EVALUACIÓN	FÓRMULA
Accuracy	$accuracy = \frac{VP + VN}{VP + VN + FP + FN}$
Precisión	$precisión = \frac{VP}{VP + FP}$
Recall	$recall = \frac{VP}{VP + FN}$
F1	$F1 = 2 * \frac{precisión * recall}{precisión + recall}$

Nota. Se presenta las fórmulas de las métricas que uso para la evaluación de los modelos y/o algoritmos utilizados.

Donde:

- VP (Verdaderos Positivos): Número de sitios web con Phishing clasificados correctamente.
- VN: Número de sitios web legítimos clasificados correctamente.
- FP: Número de sitios web legítimos clasificados erróneamente como Phishing.
- FN: Número de sitios web con Phishing clasificados erróneamente como legítimos.

Finalmente, la matriz de confusión es vital para evaluar el modelo, ofrece una visión completa de la distribución de los aciertos y errores entre clases (López y Fernández 2018). Ayuda a medir su precisión y eficacia en la tarea, siendo esencial para evaluar la calidad del modelo de aprendizaje automático. En la tabla 5

Tabla 5

Matriz de confusión para Preempt Phisher

	POSITIVOS	NEGATIVOS
POSITIVOS	Phishing Correctamente Clasificados (VP)	Legítimos Mal Clasificados (FP)
NEGATIVOS	Phishing Mal Clasificados (FN)	Legítimos Correctamente Clasificados (VN)

Nota. Se presenta la matriz de confusión para la evaluación de la extensión Preempt Phisher.

Diseño y Análisis del Sistema

En esta sección, se aborda completamente el desarrollo del sistema empleado para alcanzar el objetivo principal del proyecto, se proporciona el diseño detallado de la interfaz destinada a la creación de la extensión de Google Chrome. Esto se logra mediante la elaboración de mockups o maquetas en español, los cuales son bocetos de una interfaz de Usuario (UI) que ayuda a abordar aspectos generales de la interfaz de usuario y que cualquier parte interesada pueda crearla fácilmente (Rivero et al., 2010). El sistema de detección de phishing desarrollado recibe el nombre de "Preempt Phishing", este nombre

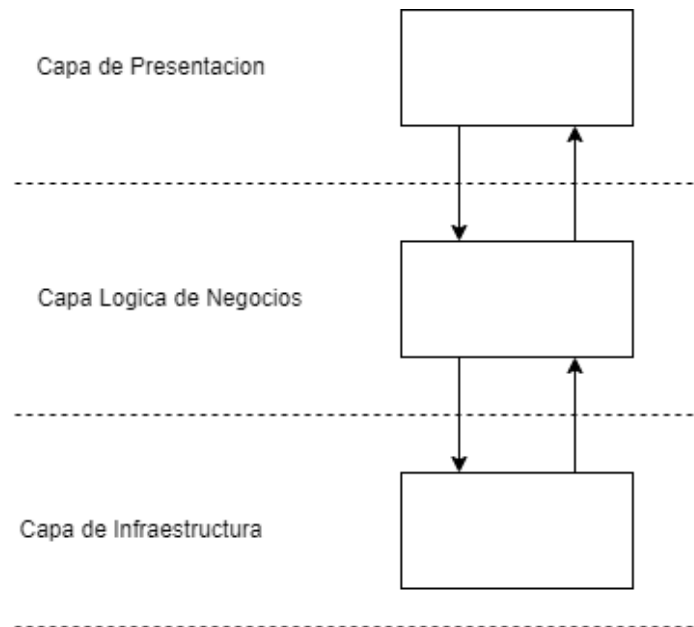
hace alusión a prevenir (Preempt) el acto de bloquear intento de robo de información a través de una estafa (Phishing).

Diseño de arquitectura. De acuerdo con (Dobrica y Niemela, 2002) en su estudio A Survey on Software Architecture Analysis Methods señala que “La arquitectura de software es un sistema que se define como la estructura o estructuras del sistema, que comprenden componentes de software, las visibles externamente propiedades de esos componentes, y las relaciones entre ellos”. Además, menciona que esta definición se centra únicamente en aspectos internos de un sistema y la mayoría de los métodos de análisis se basan en ella. El propósito de esta sección es establecer y crear la arquitectura de software empleada en la implementación del sistema propuesto, además detallar las tecnologías utilizadas para su desarrollo.

Diagrama de la arquitectura lógica. La gestión de aplicaciones en capas es de gran importancia, ya que permite separar los archivos y componentes de la aplicación, lo que conduce a un código más sencillo de mantener y reutilizar. En el caso de la extensión de Google Chrome que se está desarrollando, se seguirá el modelo de 3 capas. La presentación de este modelo se muestra en la figura 1, donde se identifican tres capas distintas: la capa de presentación, la capa lógica del negocio y la capa de infraestructura. Estas capas interactúan entre sí para asegurar el correcto funcionamiento de la aplicación. Gracias a esta organización en capas, se logra una estructura más clara y eficiente en el desarrollo del proyecto.

Figura 1

Diagrama de la arquitectura lógica del sistema



Definición de las tecnologías

Una tecnología de desarrollo de software se refiere a un programa informático utilizado para crear diversos tipos de aplicaciones. Existen numerosas tecnologías disponibles para el desarrollo de software, sin embargo, la elección de la tecnología adecuada depende del tipo de proyecto de desarrollo que se esté llevando a cabo.

En la figura 2, se muestran las tecnologías que se emplearon en cada capa del modelo descrito en la figura 1, se realiza con el objetivo de brindar una mejor comprensión sobre la función de cada capa mencionada previamente

Capa de presentación:

La capa de presentación cumple la función de facilitar la interacción entre el usuario y la aplicación. Su responsabilidad radica en recolectar los datos generados durante esta interacción y enviarlos a las demás capas para su procesamiento y visualización. A continuación, se detallan las tecnologías que se utilizó en esta capa:

- Html (Lenguaje de Marcas de Hipertexto)
- Css (Hojas de estilo en cascada)
- JS (JavaScript)

Capa lógica del negocio:

La capa lógica del negocio engloba los programas responsables de recibir las solicitudes provenientes de la capa de presentación. Posteriormente, envía estas solicitudes a la capa de infraestructura para realizar la predicción correspondiente. Una vez que los resultados son devueltos, esta capa se encarga de enviar la respuesta de vuelta a la capa de presentación para su presentación al usuario. Para la implementación de esta capa se utilizará la tecnología:

- **JavaScript** (JS): Un lenguaje de programación de alto nivel que es compatible con diversos sistemas operativos y que permite el desarrollo de una amplia gama de aplicaciones.

Capa de infraestructura

En esta capa se incluyen los componentes requeridos para interactuar con la API propuestas en este proyecto y el modelo de Machine Learning entrenado. Estos elementos principales de la capa son: el modelo entrenado que se guarda como archivo SAV y un servicio que permite realizar la predicción de un sitio web, enviado a través de la extensión de Google Chrome. A continuación, se detalla las tecnologías que se utiliza para desarrollar el servicio:

- Python
- Flask

Figura 2

Diagrama de la arquitectura lógica del sistema con las tecnologías a usar





Capa de Presentación	Tecnologías		<ul style="list-style-type: none"> - HTML - CSS - JavaScript
	Front-End		
Capa Lógica de Negocio	Back-End		<ul style="list-style-type: none"> - JavaScript
Capa de Infraestructura	Modelo		<ul style="list-style-type: none"> - SAV File
			
	Servicio		<ul style="list-style-type: none"> - Python - Flask
			

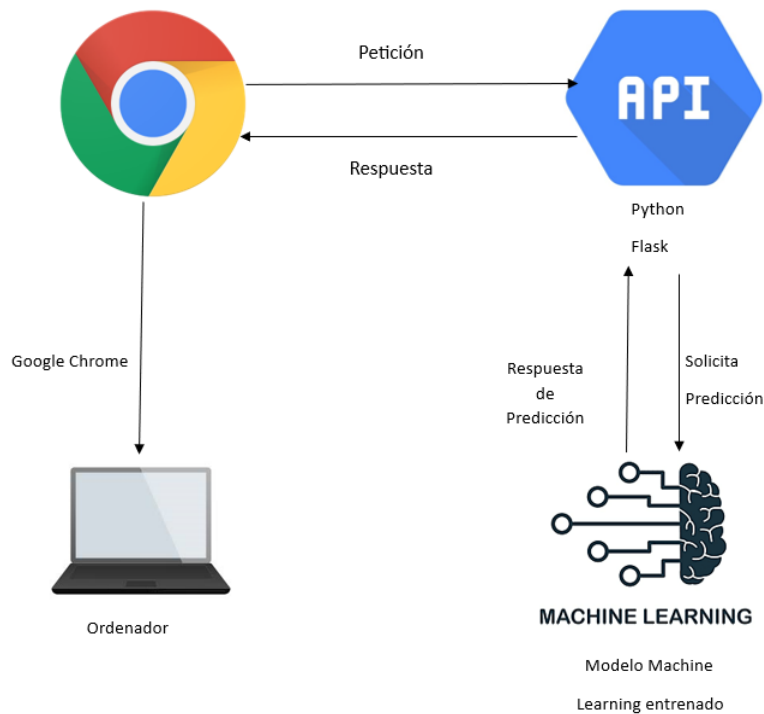
Diagrama de la arquitectura física del sistema En la figura 3 se muestra la estructura física que se emplea para desarrollar la extensión de Google Chrome llamada "Preempt Phishing". Esta arquitectura permite el acceso al sistema desde cualquier computadora que cuente con el navegador Google Chrome instalado, siempre y cuando sea una versión posterior a la 4. De esta manera, los usuarios pueden beneficiarse de la funcionalidad de la extensión y disfrutar de una experiencia segura al navegar en línea.

Cuando el usuario está navegando por internet y activa la extensión, se procederá a enviar la URL del sitio web actual. Esta información será recibida por la API, la cual se encargará de extraer todas las características relevantes tanto de la URL como del propio sitio web. Estas características se utilizarán como entrada para el modelo de Machine Learning previamente entrenado. Posteriormente, el modelo realizará su evaluación y emitirá un veredicto determinando si el sitio web es legítimo o si contiene elementos de

phishing. Finalmente, la API envía la respuesta correspondiente a la extensión, permitiendo que el usuario pueda tomar decisiones más informadas y seguras al interactuar con el sitio web en cuestión.

Figura 3

Diagrama de arquitectura física del sistema.



Nota. Se presenta el diagrama de la arquitectura física que se utilizó en el desarrollo del sistema.

Análisis del sistema

Según las pausas de la metodología Scrum, para definir los requisitos del sistema, se utilizan las Historias de Usuario (HU) las mismas que describen las necesidades del cliente en un lenguaje comprensible y sencillo. Los miembros del equipo se involucran en esta sección para entender y trabajar en la implementación estos mismos son: el propietario del proyecto o usuario del sistema (product owner), el encargado de gestionar y controlar el equipo de desarrollo (scrum máster) y el equipo de desarrollo (development team) (Kurnia et al., 2018). La asignación de los diferentes roles de Scrum se llevó a cabo según lo

especificado en la tabla 6. En dicha tabla, se detallan el nombre de cada rol y su respectivo integrante en el proyecto que ocupa esa posición.

Tabla 6

Rol de Scrum designados

N°	Rol	Integrante	Descripción
01	Scrum Master	Jorge Andres Armas Ruales	Líder del Equipo de Scrum
02	Product Owner	Dr. José Luis Carrillo Medina	Representante de las Partes Interesadas
03	Team Development	Anthony Alexander Simbaña Shuguli Jorge Andres Armas Ruales	Desarrollo y Diseño de la aplicación
04	Auditora	Ing. María Alexandra Corral Diaz	Auditora de los procesos de desarrollo

Nota. Se presenta los roles de cada uno de los integrantes del proyecto.

Análisis de Requisitos

El Scrum Master del proyecto lleva a cabo la reunión inicial, en la cual participan el usuario del sistema y el equipo de desarrollo. Al realizar la reunión se han detallado los requisitos del sistema los cuales han sido el resultado de la aplicación de técnicas de obtención de requisitos como las entrevistas, encuestas y análisis de documentos (Wiegers y Beatty, J. 2013). Se utilizó la técnica de lluvia de ideas junto con el equipo de trabajo para definir los requisitos en base a un usuario final, los cuales fueron plasmados en una historia de usuario donde se presenta un contexto de las características que tendrá el sistema.

La historia de usuario es una técnica utilizada en la gestión de proyectos y desarrollo de software ágil para describir una funcionalidad amplia y compleja que abarca los requerimientos (Cohn. 2004).

En la tabla 7, se presenta la **HU** redactada, donde se especifica el rol del usuario final del sistema, las características y/o funcionalidades requeridas, y la razón por la cual deben ser implementadas.

Tabla 7

Historias de usuario

ID	Nombre	Rol	Característica/ Funcionalidad	Razon / Resultado
1	H.U. 01	Como Usuario	Deseo una extensión para el navegador Google Chrome que pueda proporcionarme información acerca de la presencia de phishing en un sitio web. Además, quiero incorporar métodos específicos de prevención de ataques Phishing mediante IPS, detección basada en firmas y finalmente basada en anomalías	Para disponer de un medio para verificar si estoy visitando un sitio web seguro mientras navego en Internet a través de Google Chrome. Disponer de un método específico de prevención de ataques phishing, mediante IPS, detección basada en firmas y anomalías con el objetivo aumentar protección y reducir la probabilidad de ser víctimas de ataques.

Nota. Se presenta la historia de usuario una vez redactada, donde se especifica el rol.

En la tabla 8, se presenta la historia de usuario propuesta que será desarrollada a lo largo del proyecto. Está acompañada de una estimación de tiempo en días, la fecha de inicio, la fecha de finalización y el número de Sprint al que pertenece la historia de usuario mencionada.

Tabla 8

Product Backlog del proyecto

Historia de Usuario	Nombre	Estimación (días)	Fecha de Inicio	Fecha Fin	N° de Sprint
1	H.U. 01	23	06/07/2023	13/08/2023	04

Nota. Se presenta el product Backlog del proyecto, con su respectiva Fecha de Inicio y Fecha Fin.

Lista de Tareas

Scrum ayuda al trabajo iterativo, no requiere de esfuerzo sobre cómo se hacen las cosas y en qué orden se realizarán, además la existencia del Sprint Backlog el cual funciona como una lista de tareas en el sprint. Una característica del producto se divide en tareas más pequeñas (Popli y Chauhan, N. 2011) como se muestra en la tabla 9, se presenta la lista de tareas que serán desarrolladas a lo largo del proyecto, además de la especificación del rol de los desarrolladores, características y/o funciones requeridas, y la razón por la cual serán implementadas basadas en la historia de usuario.

Tabla 9

Lista de Tareas

ID	Nombre	Característica/Funcionalidad	Razon/Resultado
1	L.T 01	La extensión utiliza el algoritmo y/o modelo más efectivo de Machine Learning para detectar el phishing	Para que la extensión logre hacer predicciones con alta precisión.

ID	Nombre	Característica/Funcionalidad	Razon/Resultado
		en sitios web.	
2	L.T 02	Se crea el dataset que incluye características con Indicadores de Compromiso que posibiliten la identificación de sitios web con phishing de los sitios legítimos.	Para entrenar el modelo de Machine Learning
3	L.T. 03	El modelo de Machine Learning se aloja en un servidor para que pueda llevar a cabo las predicciones mediante un servicio.	Para crear un servicio que sea accesible y utilizable en diversas aplicaciones

Nota. Se presenta las tareas realizadas que se realizó junto con su funcionalidad.

En la tabla 10 se presenta la lista de tareas propuestas desarrolladas en el proyecto. Cada lista de tarea viene acompañada de una estimación de tiempo en días, la fecha de inicio, la fecha de finalización y el número de Sprint al que pertenece cada historia de usuario especificada

Tabla 10

Product Backlog de la Lista de Tareas

Lista de Tareas	Nombre	Estimación (días)	Fecha de Inicio	Fecha Fin	Nº de Sprint
1	L.T. 01	22	04/05/2023	25/05/2023	01
2	L.T. 02	21	26/05/2023	15/06/2023	02
3	L.T. 03	21	16/06/2023	05/07/2023	03

Nota. Se presenta el Product Backlog con la Lista Tareas realizadas junto con su fecha de inicio y Fecha fin.

Desarrollo Metodológico

Herramientas y Gestores para el Desarrollo de la Aplicación

Visual Studio Code

Se utilizó Visual Studio Code para el entorno de desarrollo ya que ofrece una alta configuración para poder escribir, depurar y administrar el código, lo cual permite la facilidad de identificación y solución de problemas, aparte que ayudó a la integración con la otra herramienta que se utilizó como Git.

PythonAnywhere

PythonAnywhere es una plataforma en la nube la cual permite ejecutarlas en línea sin necesidad de configurarlas en entorno local, ejecutar el código desde cualquier acceso a internet. Esta plataforma proporcionó el espacio para alojar la API desarrollada junto con el modelo entrenado y probar su funcionamiento en otras máquinas, conectándonos solo a la dirección donde está alojada.

Postman

Postman es una herramienta para probar APIs, fue de gran ayuda para el proyecto porque involucra comunicación con un servidor web en este caso PythonAnywhere. Permite enviar solicitudes HTTP al API desarrollado y analizar las respuestas para asegurar el funcionamiento que se espera.

GitHub

GitHub es una plataforma de alojamiento de código y colaboración, fue esencial para el desarrollo de este proyecto, porque permitió el control de versiones, seguimiento de cambios en el código durante el proyecto, aparte de ayudar de una manera eficiente el trabajo colaborativo entre los miembros del equipo y gestionar sus contribuciones.

Mockups

Los mockups son representaciones visuales que muestran cómo se verá una interfaz de usuario o diseño de una aplicación o sitio web. Son utilizados como herramientas de diseño para comunicar la estructura, el diseño y la disposición de los elementos de la interfaz antes de comenzar la fase de desarrollo (Snyder C. 2003).

Los mockups serán utilizados en el proyecto para poder tener una guía con respecto a la apariencia visual y la disposición de los elementos.

Implementación de algoritmos y modelos de Machine Learning para sitios web

Phishing.

En la metodología Scrum establece que después de realizarse el Product Backlog del proyecto, donde se describen las historias de usuario unido con el número de los Sprints asignado a cada una refiriéndose a su prioridad de desarrollo, se realiza la planificación de cada Sprint, se organiza en base a su prioridad. Se crea un Sprint Backlog por cada uno de ellos, la cual servirá como guía para la ejecución según lo planificado. (Ken y Sutherland, 2020). Cabe destacar que para cumplir las reuniones según la metodología Scrum, se utilizó la plataforma Google Meet para estas reuniones virtuales. Aparte que se fueron realizando reuniones presenciales cuando se las considero importantes.

En caso del hardware utilizado para el desarrollo de este proyecto en todos los Sprint fue el sistema operativo Windows 11, un disco duro SSD de 1000 GB Kingston, Memoria RAM DDR4 de 32 GB, Tarjeta de Video MSI GEFORCE RTX 3060 12GB y finalmente un procesador Intel Core i7-13700K, 3.4GHZ de 16 CORE.

Sprint 1: Selección del mejor modelo y/o algoritmo de Machine Learning

En el inicio del primer Sprint, se establece como punto de partida la lista de tarea L.T. 01 que se detalla en la tabla 9. Dicha historia de usuario tiene como objetivo la

selección del algoritmo y/o modelo de Machine Learning más adecuado para la prevención de sitios con phishing.

Lista de Tareas detalladas. tabla 11. Se describe en detalle la Tarea L.T. 01 del sistema de prevención de phishing, llamado “Preempt Phishing”. Se establecen los responsables del desarrollo y los criterios de aceptación para la selección del mejor modelo y/o algoritmo destinado a la prevención de sitios web que tengan phishing.

Tabla 11

Tarea para la selección del Mejor Modelo/Algoritmo y Uso de Indicadores de Compromiso para la Prevención de Phishing en Sitios Web.

Lista de Tarea	
Número: L.T. 01	Usuario: Usuario de internet
Nombre de Historia: Selección del Mejor Modelo/Algoritmo para la Prevención de Phishing en Sitios Web	
Prioridad: Alta	Riesgo en desarrollo: Media
Días estimados: 22	Interacción asignada: 1
Programadores responsables: Jorge Armas, Anthony Simbaña	
Descripción:	
<ul style="list-style-type: none"> - La extensión utilizará Indicadores de Compromiso y el algoritmo y/o modelo más efectivo de Machine Learning para prevenir el phishing en sitios web 	
Actividades:	
<ul style="list-style-type: none"> - El modelo y/o algoritmo seleccionado ha sido integrado adecuadamente en la extensión de prevención de phishing. - Se ha seleccionado el modelo y/o algoritmo de Machine Learning más efectivo y con mayor precisión en la detección de phishing, basado en las pruebas y evaluaciones realizadas - La extensión ha sido probada exhaustivamente para asegurar su correcto funcionamiento y precisión en la detección de phishing. 	

Nota. Se presenta la tarea N 1, establecen los responsables del desarrollo y los criterios de aceptación para la selección del mejor modelo y/o algoritmo destinado a la prevención de sitios web que tengan phishing.

Sprint Backlog: En la tabla 12, se detallan las actividades que fueron llevadas a cabo durante el desarrollo del Sprint, junto con los responsables de cada una, las fechas planificadas para la ejecución del sprint, estimaciones de tiempo en horas para cada tarea, y el estado actual de cada actividad.

Tabla 12

Sprint Backlog 01

Sprint 01						
Fecha Inicio:			Fecha Fin:		Jornada	
04/05/2023			25/05/2023		8 horas diarias	
H. U	Tareas	Horas	Fecha Inicio	Fecha Fin	Responsable	Estado
01	Selección de artículos	24	04/05/2023	06/05/2023	Jorge Armas y Anthony Simbaña	Finalizado
01	Extracción de modelos y/o algoritmos de Machine Learning	24	07/05/2023	09/05/2023	Jorge Armas y Anthony Simbaña	Finalizado
01	Extracción de valores del accuracy respectivamente	24	10/05/2023	12/05/2023	Jorge Armas y Anthony Simbaña	Finalizado

Sprint 01

e

01	Documentación de la Frecuencia de los modelos y/o algoritmos de Machine Learning	24	13/05/2023	15/05/2023	Jorge Armas y Anthony Simbaña	Finalizado
01	Selección de modelos y/o algoritmos de Machine Learning para realización de pruebas	24	16/05/2023	18/05/2023	Jorge Armas y Anthony Simbaña	Finalizado
01	Implementación de modelos y/o algoritmos de Machine Learning seleccionados	16	19/05/2023	20/05/2023	Jorge Armas y Anthony Simbaña	Finalizado
01	Ejecución de pruebas	16	21/05/2023	22/05/2023	Jorge Armas y Anthony Simbaña	Finalizado

Sprint 01						
01	Documentación de métricas de evaluación resultantes	16	23/05/2023	24/05/2023	Jorge Armas y Anthony Simbaña	Finalizado
01	Selección de modelo y/o algoritmo de Machine Learning con mejor valor de accuracy	8	25/05/2023	25/05/2023	Jorge Armas y Anthony Simbaña	Finalizado

Nota. Se presenta el Sprint N 1, donde se las tareas realizadas, hora y fechas respectivas.

Resultados del Sprint. En esta sección se proporciona una breve explicación del proceso realizado y los resultados más relevantes obtenidos durante la ejecución del Sprint 01. Después de realizar la revisión de la literatura, como se detalla en el capítulo II se identificaron un total de 22 modelos y/o algoritmos de Machine Learning utilizada para la detección de Phishing en sitios web.

Para cada artículo se registró la precisión (accuracy) alcanzada por cada modelo para la detección de Phishing. Además, se recopiló la frecuencia de cada modelo y/o algoritmo mencionado en los artículos. Con base en este análisis, se seleccionaron los modelos y/o algoritmos que tenían más frecuente, se eligieron 8 modelos y/o algoritmos, el K-means no fue utilizado debido a su baja precisión en el análisis de datos y solo funciona como etiquetado, que serían usados. estos algoritmos se describen en la tabla 3 junto con una breve explicación

Una vez identificados los modelos y/o algoritmos de Machine Learning con los mejores valores de precisión en la detección de phishing, se procedió a implementarlos utilizando un conjunto de datos que se muestran en el Anexo 2.

En la figura 4, se presenta el código desarrollado para la incorporación de los modelos y/o algoritmos de Machine Learning, acompañado de sus correspondientes métricas de evaluación, tales como: accuracy, precision, recall y f1 score.

La tabla 13 presenta las métricas de evaluación obtenidas tras realizar las pruebas a cada uno de los modelos y/o algoritmos seleccionados. Estos resultados fueron utilizados como base para decidir qué modelo se implementará en el desarrollo del sistema, tomando en cuenta la precisión (accuracy), ya que esta medida nos indica el porcentaje de elementos clasificados correctamente tanto en la categoría positiva como negativa.

Figura 4

Implementación de modelos y/o algoritmos de Machine Learning

```
#Random Forest
from sklearn.ensemble import RandomForestClassifier
rforest_clf = RandomForestClassifier()
cross_val_scores = cross_validate(rforest_clf, X, y, cv=500, scoring = scoring)
rforest_clf_score = mean_score(cross_val_scores)
print(rforest_clf_score)
✓ 2m 9.4s Python
{'fit_time': 0.254836458369739, 'score_time': 0.0036681766510089766, 'test_accuracy': 0.8486438746438747, 'test_recall': 0.9642727272727273, 'test_precision': 0.8571927372826841, 'test_f1': 0.9060015644659782}

#Multi-Layer Perceptron classifier
from sklearn.neural_network import MLPClassifier
neural_clf=MLPClassifier(hidden_layer_sizes=(33,),max_iter=500)
cross_val_scores = cross_validate(neural_clf, X, y, cv=fold_count, scoring=scoring)
neural_clf_score = mean_score(cross_val_scores)
print(neural_clf_score)
✓ 511s Python
{'fit_time': 5.11148159589393675, 'score_time': 0.0029697179794311523, 'test_accuracy': 0.7904838989558393, 'test_recall': 0.8907293939073836, 'test_precision': 0.8474314877045728, 'test_f1': 0.8669045780362354}

#Decision Tree
from sklearn.tree import DecisionTreeClassifier
decisiontree = DecisionTreeClassifier()
cross_val_scores = cross_validate(decisiontree, X, y, cv=fold_count, scoring=scoring)
decisiontree_clf_score = mean_score(cross_val_scores)
print(decisiontree_clf_score)
✓ 0.7s Python
{'fit_time': 0.0076956111907959, 'score_time': 0.002401818427001954, 'test_accuracy': 0.8148525995359201, 'test_recall': 0.9299164383949389, 'test_precision': 0.8517490053958279, 'test_f1': 0.887405547542758}

#Ada Boost
from sklearn.ensemble import AdaBoostClassifier
adaBoost = AdaBoostClassifier()
cross_val_scores = cross_validate(adaBoost, X, y, cv=fold_count, scoring=scoring)
adaBoost_clf_score = mean_score(cross_val_scores)
print(adaBoost_clf_score)
✓ 2.7s Python
{'fit_time': 0.2109462022781372, 'score_time': 0.008301019668579102, 'test_accuracy': 0.7864665835495209, 'test_recall': 0.9096287373362983, 'test_precision': 0.8358504989126551, 'test_f1': 0.8678850365032638}

#SVM
from sklearn.svm import SVC
svc = SVC()
cross_val_scores = cross_validate(svc, X, y, cv=fold_count, scoring=scoring)
svc_clf_score = mean_score(cross_val_scores)
print(svc_clf_score)
✓ 13.4s Python
{'fit_time': 1.110419558095691, 'score_time': 0.2225175142288208, 'test_accuracy': 0.779497036222828, 'test_recall': 0.9088858590137316, 'test_precision': 0.8286935310618742, 'test_f1': 0.8667297175571548}

#Bayesiano Ingenuo
from sklearn.naive_bayes import GaussianNB
naive_bayes_classifier = GaussianNB()
cross_val_scores = cross_validate(naive_bayes_classifier, X, y, cv=fold_count, scoring=scoring)
nbc_score = mean_score(cross_val_scores)
print(nbc_score)
✓ 0.0s Python
{'fit_time': 0.004484449462890625, 'score_time': 0.0023159583936767576, 'test_accuracy': 0.2139947044363269, 'test_recall': 0.033851907938861244, 'test_precision': 0.6864795918367348, 'test_f1': 0.048967568}
```

```

from sklearn.neural_network import MLPClassifier

# Crear el clasificador MLP
mlp_clf = MLPClassifier(max_iter=500)

# Realizar la validación cruzada
cross_val_scores = cross_validate(mlp_clf, X, y, cv=10, scoring=scoring)
mlp_clf_score = mean_score(cross_val_scores)

# Imprimir el puntaje promedio de la validación cruzada
print(mlp_clf_score)
✓ 45.1s Python

time: 4.509432853565979, 'score_time': 0.0048666862297058105, 'test_accuracy': 0.8013258396305741, 'test_recall': 0.9154577464788733, 'test_precision': 0.8472000524583858, 'test_f1': 0.8763116661

#Redes Bayesianas
from sklearn.naive_bayes import GaussianNB as Bayesiano
clasificador = Bayesiano()
cross_val_scores = cross_validate(clasificador, X, y, cv=10, scoring = scoring)
rforest_clf_score = mean_score(cross_val_scores)
print(rforest_clf_score)
✓ 0.0s Python

{'fit_time': 0.005047893524169922, 'score_time': 0.0023489475250244142, 'test_accuracy': 0.28967635831130105, 'test_recall': 0.042817166154823685, 'test_precision': 0.667737003058104, 'test_f1': 0.05397518E

```

Luego de examinar los datos en la tabla 11, se seleccionó el modelo y/o algoritmo

Random Forest, debido a que presenta un valor de precisión (accuracy) de 0,84, que equivale al 84% de precisión.

Tabla 13

Resultados de pruebas modelos y/o algoritmos de Machine Learning Implementados.

Característi cas	Algoritmos Modelos	Accuracy	Recall	Precision	F1
40	Random	0,8400	0,9630	0,8572	0,9000
característic as	Forest				
	Decision	0,8132	0,9289	0,8515	0,8867
	Tree				
	Ada Boost	0,6864	0,9096	0,8358	0,8678
	SVM	0,7794	0,9088	0,8286	0,8607
	Mezcla de	0,6094	0,9113	0,8482	0,8747
	Gaussianas				
	- GMM				
	Naive Bayes	0,2139	0,0330	0,6864	0,04896

Característi cas	Algoritmos Modelos	Accuracy	Recall	Precision	F1
	Redes	0,2096	0,0428	0,6677	0,0539
	Bayesianas				
	Redes	0.8001	0,9154	0,8472	0,8763
	Multicapa:				
	MLP				

Nota. Se presenta los resultados de las pruebas realizadas con 40 características.

Sprint 02: Creación del dataset

Para el desarrollo del Sprint actual, se consideró como base la tarea L.T.02 especificada en la tabla 9, se debe de conocer y seleccionar las características que se pueden extraer de URLs principalmente se basó en Indicadores de Compromiso, en segundo lugar, se crearán escenarios específicos para identificar las características más relevantes, posteriormente se genera un conjunto de datos para entrenar al modelo elegido.

Lista de tareas detalladas. La tabla 14, presenta la Lista de Tarea L.T. 02 del sistema de detección de phishing (Preempt Phishing) de forma precisa, donde se detallan los encargados del proceso de desarrollo, así como los estándares a cumplirse para que sea considerado aceptable el dataset, el cual tendrá características seleccionadas referentes a IOC para la construcción del sistema propuesto.

Tabla 14

Tarea para la creación de un dataset

Lista de Tarea	
Número: L.T. 02	Usuario: Usuario de internet
Nombre de Historia: Creación de dataset	
Prioridad: Alta	Riesgo en desarrollo: Media

Días estimados: 21

Interacción asignada: 1

Programadores responsables: Jorge Armas, Anthony Simbaña

Descripción:

- El dataset incluirá características con Indicadores de Compromiso que posibiliten la identificación de sitios web con phishing de los sitios legítimos

Actividades:

- Las características incluidas en el dataset deben estar con una descripción clara del tipo de indicador de compromiso corresponde.
- Se llevarán a cabo pruebas usando varios escenarios, en los cuales se implementarán modelos y/o algoritmos de Machine Learning. Estos escenarios cambian por la cantidad de características asignadas a cada uno de ellos.
- Se extraerán características de IOC seleccionadas a partir de una URL.
- El dataset se construirá utilizando las características de las URL de sitios web, tanto legítimos como con phishing.

Nota. Se presenta la tarea N 2, establecen los responsables del desarrollo y los criterios de aceptación para la selección del mejor modelo y/o algoritmo destinado a la prevención de sitios web que tengan phishing.

Sprint Backlog. En la tabla 15, se detalla toda la planificación y ejecución del sprint, incluyendo las tareas realizadas, los responsables de cada una, las fechas de ejecución, las estimaciones de tiempo en horas y el estado actual de cada tarea. El sprint backlog se encuentra completamente finalizado y listo para su revisión.

Tabla 15
Sprint Backlog 02

Sprint 02						
Fecha Inicio:		Fecha Fin:		Jornada		
26/05/2023		15/06/2023		8 horas diarias		
H. U	Tareas	Horas	Fecha	Fecha Fin	Respuesta	Estado

Sprint 02						
			Inicio		ble	
02	Extracción de características de URL enfocadas en indicadores de compromiso para identificar sitios web con phishing	24	26/05/2023	29/05/2023	Jorge Armas y Anthony Simbaña	Finalizado
02	Conteo de frecuencia de cada característica extraída	16	30/05/2023	31/05/2023	Jorge Armas y Anthony Simbaña	Finalizado
02	Selección de características de indicadores de compromiso para realizar pruebas	8	1/06/2023	1/06/2023	Jorge Armas y Anthony Simbaña	Finalizado
02	Ordenar las características por relevancia	8	2/06/2023	2/06/2023	Jorge Armas y Anthony Simbaña	Finalizado
02	Creación de	16	3/06/2023	4/06/2023	Jorge	Finalizado

Sprint 02

	generación de escenarios con distintas cantidades de características para detectar sitios web con phishing				Armas y Anthony Simbaña	
02	Implementación	16	5/06/2023	6/06/2023	Jorge	Finalizado
	de los escenarios de características con los modelos y/o Algoritmos de Machine Learning				Armas y Anthony Simbaña	
02	Ejecución de	8	5/06/2023	5/06/2023	Jorge	Finalizado
	pruebas con los diferentes escenarios				Armas y Anthony Simbaña	
02	Documentación	8	6/06/2023	6/06/2023	Jorge	Finalizado
	de métricas de evaluación resultantes				Armas y Anthony Simbaña	
02	Selección de	8	7/06/2023	7/06/2023	Jorge	Finalizado
	datasets con				Armas y	

Sprint 02						
	phishing y sitios				Anthony	
	web legítimos				Simbaña	
02	Limpieza y	8	8/06/2023	8/06/2023	Jorge	Finalizado
	unión de				Armas y	
	datasets				Anthony	
	seleccionados				Simbaña	
02	Implementación	32	9/06/2023	12/06/2023	Jorge	Finalizado
	de código para				Armas y	
	la extracción de				Anthony	
	características				Simbaña	
	de una URL					
02	Implementación	16	14/06/2023	15/06/2023	Jorge	Finalizado
	de código para				Armas y	
	la creación de				Anthony	
	dataset				Simbaña	

Nota. Se presenta el Sprint N 2, donde se las tareas realizadas, hora y fechas respectivas.

Resultados del Sprint. En esta sección se proporciona una breve explicación del proceso realizado y los resultados más relevantes obtenidos durante la ejecución del Sprint 02. En un estudio previo (Castillo M y Chuquitarco K, 2023) contiene características (recursos de comprobación) extraídas de los artículos presentados en el mismo estudio y los datasets seleccionados del repositorio KAGGLE, con la información resumida en el Anexo 2. Cada característica posee una breve descripción.

En este Sprint, se realizará la implementación del código para realizar pruebas en diversos escenarios previamente seleccionados para validación. Para cada uno de estos escenarios, se agruparán 40 características específicas que servirán como recursos de comprobación 30 pertenecen a características de URL mientras que 10 se basaran en

Indicadores de Compromiso. Se agruparon 30 características con 1 Indicador de Compromiso respectivamente de lo cual se tomó en cuenta las métricas de evaluación para validar cada una de las combinaciones, una vez realizado esto se separaron las combinaciones más relevantes. Se crean 6 escenarios, la razón de los mismos es probar cual es el aporte de las características más sobresalientes, las menos relevantes y las que están en medio, así como la combinación de estos grupos, teniendo como meta determinar características que más aporten a la detección de phishing en sitios web, para lo cual los modelos se validaron de manera individual y combinada entre ellos, por ejemplo: en primer lugar se probó las 30 características de URL, la 1 y 2 característica IOC, en segundo lugar se probaron 30 características con la 4 y 6 característica IOC, en tercer lugar se probaron las 30 con la 9,10 característica IOC, luego se probaron combinando las 30 características con la 1,2 y 4 característica IOC, después se probaron combinando las 30 características con la 6,9,10 característica IOC, y finalmente se probaron combinando las 30 características con la 1,2,4,6,9,10 características IOC, como se puede evidenciar en la figura 5.

El resultado de las cuarenta características unidas fue probado en Sprint 01, específicamente en la tabla 11. Es importante destacar que las pruebas se realizaron utilizando 6 modelos y/o algoritmos de Machine Learning seleccionados y mostrados en la tabla 3.

En la tabla 16, se presenta el resultado de la ejecución del código que se muestra en la figura 5, los cuales se sometieron a las mismas métricas de evaluación específicas al inicio de este capítulo, es decir, para medir el rendimiento de los algoritmos de Machine Learning se propuso cuatro métricas: Accuracy, Precision, Recall y F1. Para la ejecución de estas pruebas se utilizó el dataset que se encuentra en el Anexo 2, que contenía diversidad de URLs legítimas, así como también, URLs con phishing.

En primer lugar, se utilizó la métrica de exactitud (accuracy) para evaluar el desempeño del modelo. Esta métrica calcula el porcentaje de sitios web correctamente clasificados como phishing o legítimos en relación con el total de datos de entrenamiento, razón por la cual se consideró como la métrica más importante de evaluación, el primer

escenario (con las 30 características y 1,2 característica IOC) se obtuvo el accuracy el valor de 0.8134 (81.34%) proveniente del algoritmo Decision Tree. En el segundo escenario (30 características y 4,6 características IOC) se tuvo como mejor accuracy 0.8046 (80,46%) proveniente del algoritmo Decision Tree.

Finalmente, en el tercer escenario (30 características y 9,10 características IOC) se tuvo como mejor accuracy 0.8280 (82,80%) proveniente del algoritmo Random Forest. Tras examinar los resultados obtenidos en cada uno de los tres escenarios evaluados, se concluye que las 30 características y 9,10 características IOC son las que más aportan en la detección de phishing en sitios web. Por otro lado, se combinaron estos escenarios, con la finalidad de conocer cuál de estas combinaciones aumentan y/o disminuyen el valor de accuracy de detección. Se inicio con el primer escenario y el segundo (30 características y 1,2 características IOC con la 4 característica IOC), en donde, se obtuvo como mejor accuracy 0,8400 (84,00%) proveniente de del algoritmo Random Forest.

Después se combinó el tercer y segundo escenario (30 características y 9,10 características IOC con la 6 característica IOC) para obtener como mejor accuracy el valor de 0,8283 (82,83%) proveniente del algoritmo Random Forest. Finalmente se combinaron los tres escenarios para obtener como mejor accuracy el valor de 0,8402 (84,02). Es importante mencionar que, los resultados de las cuarenta características se encuentran en la tabla 11.

La métrica de precisión (precision) determina el porcentaje de sitios web con phishing bien clasificados con respecto a todos los sitios web clasificados como phishing, además se realizó el mismo proceso que en la métrica accuracy, teniendo en el 1 escenario (con las 30 características y 1,2 característica IOC) se obtuvo la mejor precisión el valor de 0.8502 (85.02%) proveniente del algoritmo Decision Tree. En el segundo escenario (30 características y 4,6 características IOC) se tuvo como mejor Precisión 0.8444 (84,44%) proveniente del algoritmo Decision Tree. Finalmente, en el tercer escenario (30 características y 9,10 características IOC) se tuvo como mejor Precisión 0.9261 (92,61%) proveniente del algoritmo Ada Boost. Los resultados obtenidos en los tres escenarios

probados de manera individual, nos indica que las 30 características y 9,10 características IOC son las que más aportan en la detección de phishing en sitios web.

De igual manera, se realizaron combinaciones de estos escenarios para determinar la precisión de detección, el objetivo es aumentar o disminuir la capacidad para detectar sitios web phishing de manera precisa. Se inició con el primer escenario y el segundo (30 características y 1,2 características IOC con la 4 característica IOC), en donde, se obtuvo como mejor Precisión 0,8562 (85,63%) proveniente del algoritmo Random Forest. Después se combinó el tercer y segundo escenario (30 características y 9,10 características IOC con la 6 característica IOC) para obtener como mejor de Precisión el valor de 0,8465 (84,65%) proveniente del algoritmo Random Forest.

Finalmente se combinaron los tres escenarios para obtener como mejor accuracy el valor de 0,85 (85%). Mediante la métrica de evaluación, se ha concluido que el algoritmo Random Forest muestra resultados prometedores con un número reducido de características. Se observa que el algoritmo mejora su precisión al utilizar un mayor número de características.

Para la métrica Recall determina el porcentaje de sitios web con phishing correctamente identificados en relación con el total de ejemplos de entrenamiento de sitios web con phishing, se realizó el mismo proceso que en la métrica accuracy y Precisión, teniendo en el 1 escenario (con las 30 características y 1,2 característica IOC) se obtuvo la mejor Recall el valor de 0.9313 (93.13%) proveniente del algoritmo Decision Tree. En el segundo escenario (30 características y 4,6 características IOC) se tuvo como mejor Recall 0.9269 (92,69%) proveniente del algoritmo Decision Tree. Finalmente, en el tercer escenario (30 características y 9,10 características IOC) se tuvo como mejor Precisión 0.9607 (96,07%) proveniente del algoritmo Random Forest.

Los resultados obtenidos en los tres escenarios probados de manera individual, nos indica que las 30 características y 9,10 características IOC son las que más aportan en la detección de phishing en sitios web. Se combinaron estos escenarios para determinar cuál de las combinaciones aumentan y/o disminuyen la métrica de evaluación Recall. Se inició

con el primer escenario y el segundo (30 características y 1,2 características IOC con la 4 característica IOC), en donde, se obtuvo como mejor Recall 0,9631 (96,31%) proveniente del algoritmo Random Forest. Después se combinó el tercer y segundo escenario (30 características y 9,10 características IOC con la 6 característica IOC) para obtener como mejor Recall el valor de 0,9613 (96,13%) proveniente del algoritmo Random Forest.

Finalmente se combinaron los tres escenarios para obtener como mejor Recall el valor de 0,9636 (96,36%). Con esta métrica de evaluación, se ha concluido que el algoritmo Random Forest muestra resultados prometedores con un número reducido de características. Se observa que el algoritmo mejora su Recall al utilizar un mayor número de características.

Para la métrica F1, se utiliza para combinar las medidas de precisión y recall en un solo valor, hace que sea más fácil el comparar rendimiento, y se realizó el mismo proceso que en la métrica accuracy, Precision y Recall, teniendo en el 1 escenario (con las 30 características y 1,2 característica IOC) se obtuvo la mejor F1 el valor de 0.8873 (88.73%) proveniente del algoritmo Decision Tree. En el segundo escenario (30 características y 4,6 características IOC) se tuvo como mejor F1 0.8821 (88,21%) proveniente del algoritmo Decision Tree.

En el tercer escenario (30 características y 9,10 características IOC) se tuvo como mejor Precisión 0.8993 (89,93%) proveniente del algoritmo Random Forest. Los resultados obtenidos en los tres escenarios probados de manera individual, nos indica que las 30 características y 9,10 características IOC son las que más aportan en la detección de phishing en sitios web. Se combinaron estos escenarios para determinar cuál de las combinaciones aumentan y/o disminuyen la métrica de evaluación F1. Se inició con el primer escenario y el segundo (30 características y 1,2 características IOC con la 4 característica IOC), en donde, se obtuvo como mejor F1 0,9057 (90,57%) proveniente del algoritmo Random Forest. Después se combinó el tercer y segundo escenario (30 características y 9,10 características IOC con la 6 característica IOC) para obtener como mejor F1 el valor de 0,8996 (89,96%) proveniente del algoritmo Random Forest. Finalmente

se combinaron los tres escenarios para obtener como mejor F1 el valor de 0,9059 (90,59%). Con esta métrica de evaluación, se ha concluido que el algoritmo Random Forest muestra resultados prometedores con un número reducido de características. Se observa que el algoritmo mejora su F1 al utilizar un mayor número de características.

Figura 5

Pruebas de características con diferentes escenarios

```

Prueba con las 30 Características mas 1 y 2 IOCs relevantes

df1 = pd.read_csv("../Características/30Carac/Dataset_Legitime_AllFeatures_30x1.csv", index_col=0)
df1 = df1.drop("result", axis=1)
df2 = pd.read_csv("../Características/1Carac/Dataset_Legitime_2x1.csv", index_col=0)
df = pd.concat([df1, df2], axis=1)
X = df.drop("result", axis=1).values
X = preprocessing.scale(X)
y = df["result"].values
df.head()

Prueba con las 30 Características mas 4 y 6 IOCs relevantes

df1 = pd.read_csv("../Características/30Carac/Dataset_Legitime_AllFeatures_30x9.csv", index_col=0)
df1 = df1.drop("result", axis=1)
df2 = pd.read_csv("../Características/1Carac/Dataset_Legitime_6x1.csv", index_col=0)
df = pd.concat([df1, df2], axis=1)
X = df.drop("result", axis=1).values
X = preprocessing.scale(X)
y = df["result"].values
df.head()

Prueba con las 30 Características mas 9 y 10 IOCs relevantes

df1 = pd.read_csv("../Características/30Carac/Dataset_Legitime_AllFeatures_30x9.csv", index_col=0)
df1 = df1.drop("result", axis=1)
df2 = pd.read_csv("../Características/1Carac/Dataset_Legitime_10x1.csv", index_col=0)
df = pd.concat([df1, df2], axis=1)
X = df.drop("result", axis=1).values
X = preprocessing.scale(X)
y = df["result"].values
df.head()

Prueba con las 30 Características mas 1, 2 y 4 IOCs relevantes

df1 = pd.read_csv("../Características/30Carac/Dataset_Legitime_AllFeatures_30x1.csv", index_col=0)
df1 = df1.drop("result", axis=1)
df2 = pd.read_csv("../Características/1Carac/Dataset_Legitime_2x1.csv", index_col=0)
df3 = pd.read_csv("../Características/1Carac/Dataset_Legitime_4x1.csv", index_col=0)
df = pd.concat([df1, df2, df3], axis=1)
X = df.drop("result", axis=1).values
X = preprocessing.scale(X)
y = df["result"].values
df.head()

Prueba con las 30 Características mas 6, 9 y 10 IOCs relevantes

df1 = pd.read_csv("../Características/30Carac/Dataset_Legitime_AllFeatures_30x6.csv", index_col=0)
df1 = df1.drop("result", axis=1)
df2 = pd.read_csv("../Características/1Carac/Dataset_Legitime_9x1.csv", index_col=0)
df3 = pd.read_csv("../Características/1Carac/Dataset_Legitime_10x1.csv", index_col=0)
df = pd.concat([df1, df2, df3], axis=1)
X = df.drop("result", axis=1).values
X = preprocessing.scale(X)
y = df["result"].values
df.head()

Prueba con las 30 Características mas 1, 2, 4, 6, 9 y 10 IOCs relevantes

df1 = pd.read_csv("../Características/30Carac/Dataset_Legitime_AllFeatures_30x1.csv", index_col=0)
df1 = df1.drop("result", axis=1)
df2 = pd.read_csv("../Características/1Carac/Dataset_Legitime_2x1.csv", index_col=0)
df3 = pd.read_csv("../Características/1Carac/Dataset_Legitime_4x1.csv", index_col=0)
df4 = pd.read_csv("../Características/1Carac/Dataset_Legitime_6x1.csv", index_col=0)
df5 = pd.read_csv("../Características/1Carac/Dataset_Legitime_9x1.csv", index_col=0)
df6 = pd.read_csv("../Características/1Carac/Dataset_Legitime_10x1.csv", index_col=0)
df = pd.concat([df1, df2, df3, df4, df5, df6], axis=1)
X = df.drop("result", axis=1).values
X = preprocessing.scale(X)
y = df["result"].values
df.head()

```

Tabla 16

Resultados pruebas modelos y/o algoritmos de Machine Learning implementados en diferentes escenarios

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
1	30	Random Forest	0.8056	0.9226	0.8475	0.8810
	Características vs 1 y 2 IOC	Multi-layer	0.7935	0.9037	0.8475	0.8699
		Perceptron classifier				
		Decision Tree	0.8134	0.9313	0.8502	0.8873
		Ada Boost	0.7982	0.9173	0.8377	0.8727
		SVM	0.7774	0.9056	0.8282	0.8589
		Mezcla de Gaussianas - GMM	0.7895	0.9000	0.8447	0.8665
		Naive Bayes	0.8095	0.9262	0.8494	0.8842
		Redes Bayesianas	0.7899	0.9116	0.8385	0.8702
		Redes Multicapa: MLP	0.7834	0.9071	0.8337	0.8634

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
2	30	Random Forest	0.7974	0.9199	0.8410	0.8762
	Características vs 4 y 6 IOC	Multi-layer	0.7860	0.9085	0.8362	0.8672
		Perceptron classifier				
		Decision Tree	0.8046	0.9269	0.8444	0.8821
		Ada Boost	0.7750	0.9046	0.8283	0.8609
		SVM	0.7774	0.9056	0.8282	0.8589
		Mezcla de Gaussianas - GMM	0.7884	0.9067	0.8403	0.8616
		Naive Bayes	0.8022	0.9267	0.8422	0.8806
		Redes Bayesianas	0.7685	0.8972	0.8252	0.8548
		Redes Multicapa: MLP	0.7707	0.8913	0.8294	0.8521
3	30	Random Forest	0.8280	0.9607	0.8465	0.8993
	Características					

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
	vs 9 y 10 IOC	Multi-layer Perceptron classifier	0.7738	0.8933	0.8333	0.8563
		Decision Tree	0.7980	0.9245	0.8396	0.8781
		Ada Boost	0.7690	0.8977	0.9261	0.8562
		SVM	0.7725	0.9032	0.8247	0.8565
		Mezcla de Gaussianas - GMM	0.7886	0.9156	0.8356	0.8704
		Naive Bayes	0.7967	0.9237	0.8384	0.8773
		Redes Bayesianas	0.7700	0.8980	0.8268	0.8568
		Redes Multicapa: MLP	0.7655	0.8903	0.8261	0.8496
4	30 Características vs 1 - 2 y 4 IOC	Random Forest	0.8400	0.9631	0.8562	0.9057
		Multi-layer Perceptron classifier	0.8143	0.9304	0.8520	0.8877

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
		Decision Tree	0.8139	0.9298	0.8517	0.8863
		Ada Boost	0.7874	0.9109	0.8361	0.8688
		SVM	0.7779	0.9070	0.8278	0.8592
		Mezcla de Gaussianas - GMM	0.8034	0.9186	0.8475	0.8788
		Naive Bayes	0.8007	0.9119	0.8498	0.8761
		Redes Bayesianas	0.7835	0.9040	0.8358	0.8641
		Redes Multicapa: MLP	0.7795	0.9052	0.8306	0.8599
5	30 Características vs 6 - 9 y 10 IOC	Random Forest	0.8283	0.9613	0.8465	0.8996
		Multi-layer Perceptron classifier	0.7783	0.8997	0.8334	0.8602
		Decision Tree	0.7982	0.9247	0.8396	0.8782

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
		Ada Boost	0.7690	0.8977	0.8261	0.8562
		SVM	0.7725	0.9032	0.8249	0.8565
		Mezcla de Gaussianas - GMM	0.7804	0.8998	0.8363	0.8622
		Naive Bayes	0.7913	0.9174	0.8372	0.8726
		Redes Bayesianas	0.7603	0.8874	0.8217	0.8476
		Redes Multicapa: MLP	0.7692	0.8988	0.8235	0.8535
6	30 Características vs 1 - 2 - 4 - 6 - 9 y 10 IOC	Random Forest	0.8402	0.9636	0.8561	0.9059
		Multi-layer Perceptron classifier	0.7963	0.9102	0.8467	0.8734
		Decision Tree	0.8130	0.9286	0.8514	0.8866
		Ada Boost	0.7874	0.9109	0.8361	0.8688

Ord.	Características	Algoritmos/ Modelos	Accuracy	Recall	Precision	F1
		SVM	0.7788	0.9081	0.8283	0.8600
		Mezcla de Gaussianas - GMM	0.8044	0.9182	0.8487	0.8892
		Naive Bayes	0.8057	0.9186	0.8509	0.8805
		Redes Bayesianas	0.7796	0.8992	0.8341	0.8602
		Redes Multicapa: MLP	0.7857	0.9132	0.8328	0.8658

Nota. Se presenta pruebas modelos y/o algoritmos de Machine Learning implementados en diferentes escenarios.

En la tabla 17, se presentan los modelos y/o algoritmos de Machine Learning que obtuvieron los mejores resultados en las métricas aplicadas. Para la métrica Accuracy, se destaca en todos los escenarios propuestos, el algoritmo Random Forest, logra el mayor porcentaje de detección precisa de sitios web con phishing y legítimos en relación con todos los datos de entrenamiento. Con respecto a la métrica Precision, se puede observar que el algoritmo Decision Tree obtiene los mejores resultados en los dos primeros escenarios, mientras que en los demás escenarios el algoritmo Random Forest predomina. En cuanto a la métrica Recall, nuevamente, se destaca el algoritmo Random Forest, se obtiene el mayor porcentaje de detección de Sitios Web con phishing y legítimos. Finalmente, en la métrica F1, se destaca el algoritmo Random Forest, se obtiene el mayor porcentaje de detección de Sitios Web dando como mayor grado de precisión el 6 escenario.

De acuerdo, a la revisión realizada se encontró el porcentaje promedio usando Indicadores de Compromiso obtenido en el artículo de (Noor et al., 2019) el cual consta de 83%. Con las pruebas realizadas al modelo propuesto en este trabajo se obtuvo un 84.0% de Accuracy más alto usando las 40 características seleccionadas, con este valor se puede indicar que se obtuvieron porcentajes dentro del rango establecido según la literatura científica revisada.

Con los porcentajes obtenidos se puede indicar que mientras más características se combinan o integran los resultados en la predicción son más altos y significativos.

En general, se observa que el algoritmo Random Forest es uno de los mejores en la mayoría de las métricas y escenarios probados, mostrando un rendimiento consistente en la detección de sitios web con phishing. Sin embargo, en algunos casos, otros algoritmos como Decision Tree y Ada Boost también alcanzan resultados destacados en ciertas métricas y escenarios específicos.

Tabla 17

Ganador de cada escenario

Ord	Escenario / Métrica	Accuracy	Precision	Recall	F1
1	Escenario (30 Características y 1 y 2 IOC)	Decision Tree (81,34%)	Decision Tree (93,13%)	Random Forest (93,13%)	Decision Tree (88,73%)
2	Escenario (30 Características y 4 y 6 IOC)	Decision Tree (80,46%)	Decision Tree (92,69%)	Decision Tree (84,44%)	Decision Tree (88,21%)
3	Escenario (30 Características y 9 y 10 IOC)	Random Forest	Random Forest	Ada Boost	Random Forest

Ord	Escenario / Métrica	Accuracy	Precision	Recall	F1
		(82,80%)	(96,07%)	(92,61%)	(89,93%)
4	Escenario (30 Características y 1 y 2 y 4 IOC)	Random Forest	Random Forest	Random Forest	Random Forest
		(84,00%)	(96,31%)	(85,62%)	(90,57%)
5	Escenario (30 Características y 9 y 10 y 6 IOC)	Random Forest	Random Forest	Random Forest	Random Forest
		(82,83%)	(96,13%)	(84,65%)	(88,96%)
6	Escenario (30 Características vs 1 y 2 y 4 y 6 y 10 y 9 IOC)	Random Forest	Random Forest	Random Forest	Random Forest
		(84,02%)	(96,36%)	(85,61%)	(90,59%)

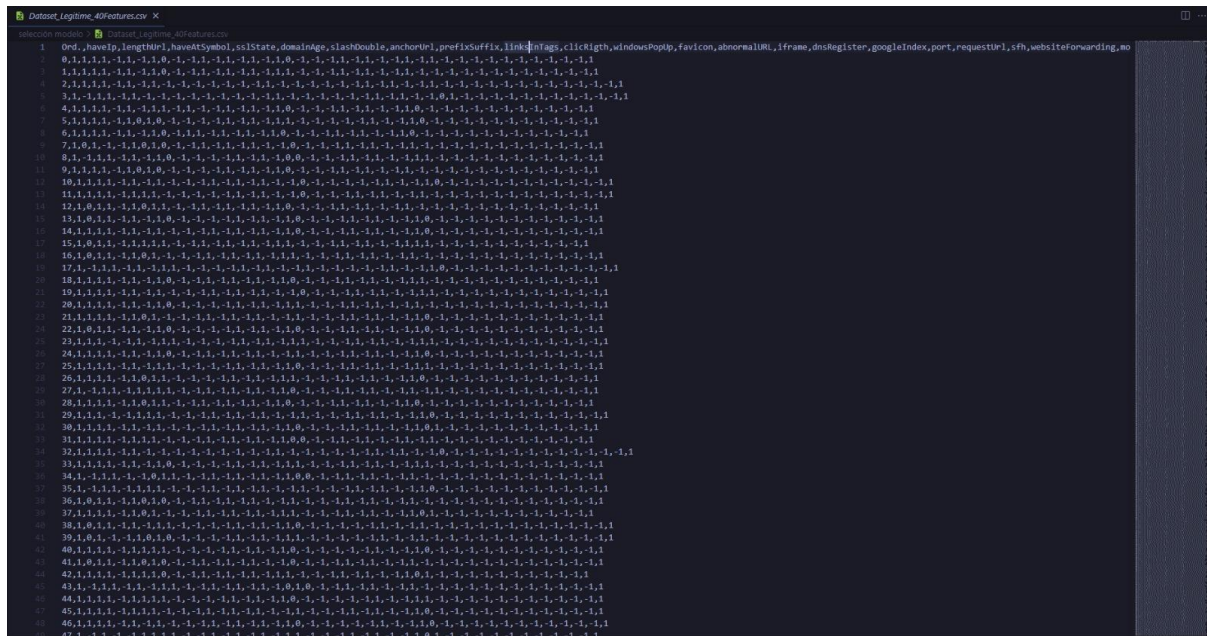
Nota. Se presenta el ganador de cada escenario.

Creación del DataSet

Para la creación del dataset requerido en Tarea L.T .02, se llevó a cabo una implementación de un código específico para extraer las 40 características seleccionadas desde una URL. El código es esencial para la creación del dataset. Cada una de las características se implementó como un método dentro de una clase, y cada método devuelve un valor numérico que representa el tipo de sitio web identificado es legítimo o no. El valor "1" indica que el sitio web es legítimo libre de Phishing y el valor "-1" indica que se trata de un sitio web con phishing. En la Figura 6, se muestra la ejecución del código para extraer características a partir de una URL, el mismo que muestra la URL enviada (<https://es-la.facebook.com/login>), la respuesta del sitio web (<Response [200]>) y un arreglo que contiene las 40 características obtenida de la URL enviada.

Figura 6

Extracción de Características del sitio web



Sprint 03: Creación de la API

En este Sprint, se utilizó como punto de partida la Tarea L.T 03, detallada en la tabla 9. El proceso de desarrollo constó tres etapas principales: entrenar el modelo y/o algoritmo de Machine Learning utilizando el conjunto de datos previamente creado, se guarda el modelo entrenado y finalmente, se implementó una API con las rutas necesarias para permitir la realización de predicciones mediante una UR.

Lista de Tarea. tabla 18, se describe en detalle la Tarea L.T. 03 del sistema de prevención de phishing, llamado “Preempt Phishing”. Se establecen los responsables del desarrollo y las actividades para la creación de la API

Tabla 18

Tarea para la creación de la API

Lista de Tarea	
Número: L.T. 03	Usuario: Usuario de internet
Nombre de Historia: Creación de la API	
Prioridad: Alta	Riesgo en desarrollo: Media

Días estimados: 21 **Interacción asignada:** 1

Programadores responsables: Jorge Armas, Anthony Simbaña

Descripción:

- El modelo de Machine Learning se aloja en un servidor para que pueda llevar a cabo las predicciones mediante un servicio.
-

Actividades:

- El servidor debe estar disponible y en funcionamiento para recibir solicitudes de predicción
 - El servicio debe estar correctamente integrado con la extensión de Google Chrome para que los usuarios puedan acceder a él de manera sencilla.
 - Se debe realizar una prueba exhaustiva del servicio para verificar su estabilidad y rendimiento bajo diferentes condiciones de carga.
-

Nota. Se presenta la tarea N 3, establecen los responsables del desarrollo y los criterios de aceptación para la selección del mejor modelo y/o algoritmo destinado a la prevención de sitios web que tengan phishing.

Sprint Backlog: En la Tabla 19, se detallan las actividades que fueron llevadas a cabo durante el desarrollo del Sprint, junto con los responsables de cada una, las fechas planificadas para la ejecución del sprint, estimaciones de tiempo en horas para cada tarea, y el estado actual de cada actividad.

Tabla 19

Sprint Backlog 03

Sprint 03						
Fecha Inicio:			Fecha Fin:		Jornada	
16/06/2023			05/07/2023		8 horas diarias	
H. U	Tareas	Horas	Fecha Inicio	Fecha Fin	Responsable	Estado
03	Guardar el	16	16/06/2023	17/06/2023	Jorge	Finalizado

Sprint 03

	modelo de Machine Learning entrenado				Armas y Anthony Simbaña	
03	Preparación del entorno local	16	18/06/2023	19/06/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Desarrollar API	16	20/06/2023	21/06/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Pruebas de API	16	22/06/2023	23/06/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Búsqueda de servidores	16	24/06/2023	25/06/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Cotización y selección de un servidor	16	26/06/2023	27/06/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Contratación de	16	28/06/2023	29/06/2023	Jorge	Finalizado

Sprint 03

	un plan básico del servidor seleccionado				Armas y Anthony Simbaña	
03	Preparación del entorno en el servidor	16	30/06/2023	31/06/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Subida de archivos	8	01/07/2023	01/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Despliegue de aplicación	8	02/07/2023	02/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Pruebas de API en el servidor	16	03/07/2023	04/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
03	Configuración de CORS	8	05/07/2023	05/07/2023	Jorge Armas y	Finalizado

Sprint 03

Anthony

Simbaña

Nota. Se presenta el Sprint N 3, donde se las tareas realizadas, hora y fechas respectivas.

Resultados del Sprint. En esta sección, se presenta una explicación del proceso que se llevó a cabo y los resultados obtenidos durante la ejecución del sprint y una vez finalizado. Al inicio se procedió a entrenar el modelo y/o algoritmo de Machine Learning selección al finalizar el Sprint 01, el cual fue el Random Forest. Durante el entrenamiento, se obtuvieron los siguientes resultados para las métricas de evaluación accuracy, precision, recall y f1. El modelo se guarda una vez entrenado completamente, tal como se muestra en la figura 8.

Figura 8

Modelo entrenado y guardado

```
#Guardar el modelo entrenado
import joblib
joblib.dump(randomForest, "randomForest-model-40.sav")
✓ 0.1s Python
['randomForest-model-40.sav']
```

```
#Guardar el modelo entrenado
import joblib
joblib.dump(randomForest, "randomForest-model.sav")
✓ 0.0s Python
['randomForest-model.sav']
```

En la arquitectura de la aplicación, se planificó utilizar Python como tecnología cliente, lo que llevó a seleccionar PythonAnywhere como el servicio de alojamiento web para desplegar la API desarrollada. PythonAnywhere está basado en el lenguaje de programación Python y el framework flask.

La figura 9, muestra los archivos esenciales para el funcionamiento de la API, incluyendo “app.py”, la cual contiene las rutas y el código necesario para las predicciones “featureExtraction.py”, que contiene el código utilizado para la extracción de características de un sitio web, finalmente, el archivo “randomForest-mode.sav”, donde se guarda el

modelo de Machine Learning seleccionado al finalizar el Sprint 01 el cual fue entrenado durante el mismo Sprint.

Figura 9

API alojado al Servidor

The screenshot displays the PythonAnywhere web interface. At the top, there is a navigation bar with links for 'Send feedback', 'Forums', 'Help', 'Blog', 'Account', and 'Log out'. Below this, the PythonAnywhere logo and 'by ANACONDA' are visible. A warning message states: 'Warning You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one here.' The main area shows the current directory path '/home/george246/' and a 'Files' section. The 'Files' section contains a table of files:

File Name	Actions	Created	Size
app.py	Download, Delete	2023-06-26 17:45	3.1 KB
featureExtraction.py	Download, Delete	2023-06-08 18:07	19.2 KB
randomForest-model.sav	Download, Delete	2023-06-08 18:32	25.2 MB

Below the file list, there is an 'Upload a file' button with a note '100MiB maximum size'. The footer contains the copyright information: 'Copyright © 2011-2023 PythonAnywhere LLP – Terms – Privacy & Cookies'.

En la figura 10, se presentan las pruebas de validación de la API desarrollada, la cual está alojada en el servidor. Estas pruebas se utilizaron para comprobar su funcionalidad, y se enviaron dos URLs de prueba. La primera URL corresponde al sitio web de YouTube (<https://www.youtube.com>), que retorna el valor de 1, mostrando que se trata de un sitio web legítimo. La segunda URL utilizada corresponde al sitio web con phishing (<http://correos-pagos.com/>), y en este caso, la API retorna el valor de -1, confirmado la presencia de phishing en el sitio.

Figura 10

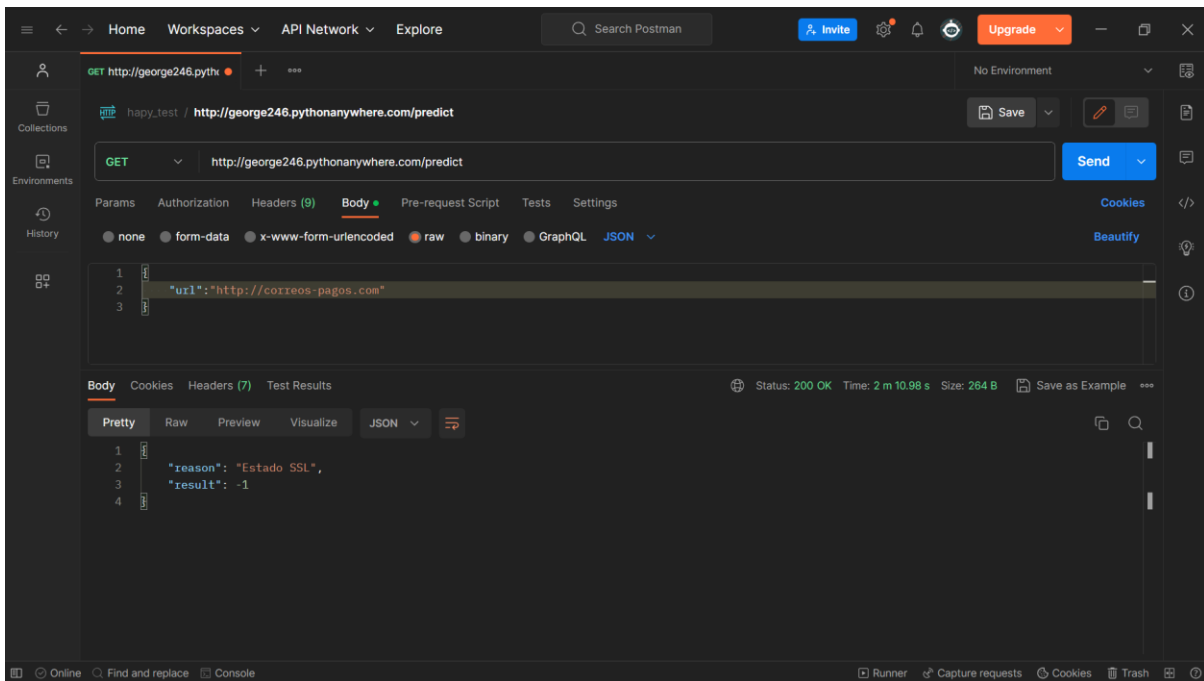
Predicción de sitios web utilizando la API desarrollada

The screenshot shows the Postman interface with a GET request to `http://george246.pythonanywhere.com/predict`. The response status is 200 OK, with a time of 224 ms and a size of 312 B. The response body is displayed in the 'Body' tab, showing a JSON object with a single key-value pair: `"mensaje": "Bienvenido, ingresa a la ruta predict para empezar la predicci3n"`.

```
1 {
2   "mensaje": "Bienvenido, ingresa a la ruta predict para empezar la predicci3n"
3 }
```

The screenshot shows the Postman interface with a GET request to `http://george246.pythonanywhere.com/predict`. The response status is 200 OK, with a time of 2 m 10.26 s and a size of 270 B. The response body is displayed in the 'Body' tab, showing a JSON object with two key-value pairs: `"reason": "Links en los Tags"` and `"result": 1`.

```
1 {
2   "reason": "Links en los Tags",
3   "result": 1
4 }
```



Desarrollo de la extensión para Google Chrome

En esta sección se detalla el Sprint dedicado al desarrollo de la extensión de Google Chrome, que constituye el objetivo central de este proyecto. Conforme a la arquitectura del sistema previamente establecida, se utilizan las tecnologías HTML, CSS y JavaScript para la creación de la aplicación Preempt Phishing.

Sprint 04: Desarrollo de la extensión de Google Chrome

Para llevar a cabo el presente Sprint, se usó la Historia de Usuario H.U.01 que se encuentra detallada en la tabla 7. En primer lugar, se empezó el desarrollo inicial de la extensión para Google Chrome utilizando HTML, en segundo lugar, se procede a aplicar el diseño a la estructura básica utilizando CSS3, finalmente se desarrolla un script en JavaScript que se embarga de consumir un servicio web destinado a la detección de sitios web con phishing. El servicio hará uso de los modelos y/o algoritmos de Machine Learning desplegados anteriormente.

Historia de Usuario detallada. La tabla 20, presenta la Historia de Usuario H.U.01 del sistema de detección de phishing (Preempt Phishing), donde se especifica los

responsables del desarrollo, además de los criterios de aceptación para la creación de la extensión.

Tabla 20

Historia de usuario para el desarrollo de la extensión de Google Chrome

Historia de Usuario	
Número: H.U. 04	Usuario: Usuario de internet
Nombre de Historia: Desarrollo de la extensión Google Chrome	
Prioridad: Alta	Riesgo en desarrollo: Media
Días estimados: 39	Interacción asignada: 1
Programadores responsables: Jorge Armas, Anthony Simbaña	
Descripción:	
<ul style="list-style-type: none"> - Como usuario deseo una extensión para el navegador Google Chrome que pueda proporcionarme información acerca de la presencia de phishing en un sitio web. - Implementación de Métodos de Prevención de Ataques Phishing mediante IPS, 	
Criterios de Aceptación:	
<ul style="list-style-type: none"> - Se desarrollará la extensión para que sea compatible con el navegador Google Chrome - La extensión consume un servicio web para la predicción - Deberá capturar la URL del sitio web donde se encuentra el usuario y realizar la predicción - Mostrará un mensaje si el sitio contiene o no phishing - Se debe implementar un sistema de notificación dando a conocer las causas del bloqueo de la página. - El sistema establece un cierre sobre la página después de que se muestra la notificación. - Se debe realizar una revisión y pruebas de los métodos de prevención y 	

garantizar su correcto funcionamiento y eficacia.

Nota. Se presenta la Historia de Usuario, establecen los responsables del desarrollo y los criterios de aceptación para la selección del mejor modelo y/o algoritmo destinado a la prevención de sitios web que tengan phishing.

Sprint Backlog. En la tabla 21, se detallan las tareas realizadas para completar el desarrollo del sprint, así como los responsables de cada tarea, las fechas planificadas para su ejecución, las estimaciones de tiempo en horas y el estado actual de cada tarea. Es importante mencionar que el sprint backlog ha sido completado y se encuentra listo para su revisión.

Tabla 21

Sprint Backlog 04

Sprint 04						
Fecha Inicio:		Fecha Fin:		Jornada		
06/07/2023		13/08/2023		8 horas diarias		
H. U	Tareas	Horas	Fecha Inicio	Fecha Fin	Responsable	Estado
04	Creación del archivo de configuración de la extensión	16	06/07/2023	07/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Creación de diseño visualmente atractivo y fácil para el uso de la extensión	16	08/07/2023	9/07/2023	Jorge Armas y Anthony Simbaña	Finalizado

Sprint 04

04	Codificación de HTML	16	10/07/2023	11/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Utilizar codificación de estilos CSS3 y mejorar la apariencia de interfaz gráfica.	16	12/07/2023	13/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Codificación para la obtención del sitio web actual	16	14/07/2023	15/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Codificación del script para el consumo de la API	32	16/07/2023	19/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Pruebas de funcionamiento unitarias de extensión	16	20/07/2023	21/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Corrección de errores	32	21/07/2023	24/07/2023	Jorge Armas y Anthony Simbaña	Finalizado

Sprint 04						
04	Optimización del rendimiento	16	24/07/2023	25/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Documentación	16	25/07/2023	26/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Empaquetado y distribución	16	27/07/2023	28/07/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Análisis de documentación respectiva en métodos de prevención de ataques basado IPS, firmas y anomalías	32	29/07/2023	01/08/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Implementación de código para la prevención de ataques basado en IPS.	32	02/08/2023	05/08/2023	Jorge Armas y Anthony Simbaña	Finalizado
04	Implementación	16	06/08/2023	07/08/2023	Jorge	Finalizado

Sprint 04

	de código para la prevención de ataques basado en Firmas				Armas y Anthony Simbaña	
04	Implementación	16	08/08/2023	09/08/2023	Jorge	Finalizado
	de código para la prevención de ataques basado en anomalías				Armas y Anthony Simbaña	
04	Desarrollo de	16	10/08/2023	11/08/2023	Jorge	Finalizado
	código para la obtención de la característica detectada en el servidor alojado en la nube				Armas y Anthony Simbaña	
04	Pruebas y	16	12/08/2023	13/08/2023	Jorge	Finalizado
	Corrección de errores				Armas y Anthony Simbaña	

Nota. Se presenta el Sprint N 4, donde se las tareas realizadas, hora y fechas respectivas.

Resultados del Sprint. En esta sección, se presenta la explicación del proceso llevado a cabo y los resultados más destacados durante la ejecución del sprint y al

finalizarlo. El primer paso fue crear el archivo “manifest. json” el cual tenía la configuración como el nombre, descripción, icono, versión y permisos necesarios para la extensión.

Luego se procedió a generar los directorios destinados a contener estilos, imágenes y scripts. Luego se llevó a cabo la construcción de la estructura y diseño del aplicativo. Por último, se implementó el script correspondiente para realizar predicciones de sitios web con phishing.

En la figura 11: Se presenta el resultado final del Sprint, mostrando la extensión de Google Chrome en distintos escenarios. En la (figura 11a) se puede observar la extensión en funcionamiento con un sitio web legítimo. En la (figura 11b), la extensión está consumiendo el servicio web desarrollado. Por último, en la (figura 11c), se muestra la extensión detectando un sitio web con phishing.

Figura 11

Extensión de Google Chrome desarrollada.

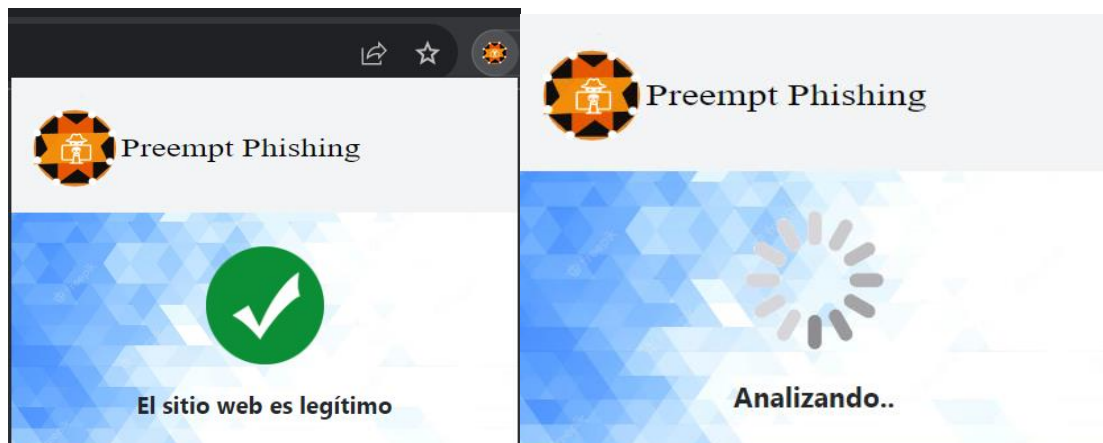


Fig 11.a Extensión

Fig 11.b Extensión analizando



Fig 11.c Extension detectando web con phishing

Con los resultados obtenidos se implementó en el código para al momento de detectar un ataque phishing de una página, que el sistema envíe una notificación al usuario sobre la causa del ataque phishing como se muestra en la figura 12, además se presenta una pantalla donde se muestra un bloqueo a la URL ingresada como se muestra en la figura 13.

Figura 12

Notificación enviada al momento de la detección de phishing en el sitio web.

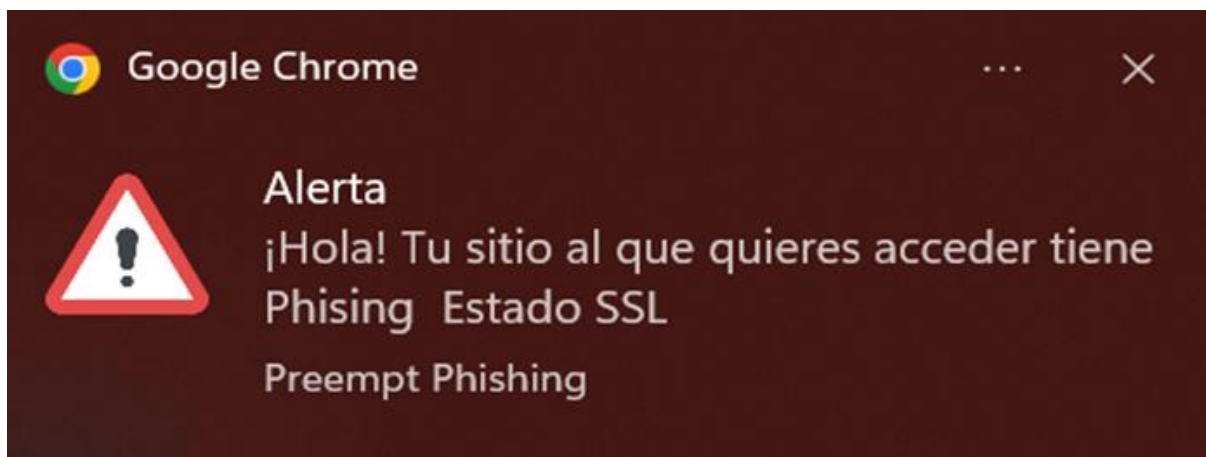


Figura 13

Bloqueo de pantalla al sitio web detectado como malicioso.



Ejecución

La ejecución se tomó en cuenta dependiendo de la realización con respecto a cada uno de los Sprints como primer punto la elección del mejor algoritmo para las predicciones de sitios web con phishing o legítimos en la cual se eligieron 6 modelos y/o algoritmos de Machine Learning como se muestra en la figura 14.

Figura 14

Modelos/Algoritmos de Machine Learning.

```
#Random Forest
from sklearn.ensemble import RandomForestClassifier
rforest_clf = RandomForestClassifier()

#Multi-layer Perceptron classifier
from sklearn.neural_network import MLPClassifier
neural_clf=MLPClassifier(hidden_layer_sizes=(33,),max_iter=500)

#Decision Tree
from sklearn.tree import DecisionTreeClassifier
decisionTree = DecisionTreeClassifier()

# Ada Boost
from sklearn.ensemble import AdaBoostClassifier
adaBoost = AdaBoostClassifier()

#SVM
from sklearn.svm import SVC
svc = SVC()

#Bagging Classifier Random Forest
from sklearn.ensemble import BaggingClassifier
bagging = BaggingClassifier(RandomForestClassifier())
```

Cada uno fue tomado de una revisión de la literatura antes mencionada. Una vez se mencionaron los modelos a usar, se tomó la iniciativa de la creación de un dataset el cual servirá para entrenar al modelo de Machine Learning. Este mismo dataset contendrá las

características de los Indicadores de Compromiso los cuales ayudaran de una manera exponencial en marketing. Esto se puede evidenciar en la figura 15.

Figura 15

Algoritmo de 40 características para generar Dataset

```
# LIBRERIAS A UTILIZAR
import pandas as pd
# Clase website creado
import feature_extraction40

#IMPORTAMOS LOS DATOS
try:
    datasetUrl = pd.read_csv('data_1.csv')
except pd.errors.ParserError as e:
    print(f'Error al leer el archivo CSV: {e}')
    datasetUrl = pd.DataFrame()

dataframeUrl = pd.DataFrame(data = datasetUrl)
websiteList = []
listDoubt = []
#Website legitimos y con phishing
n = len(dataframeUrl)
for i in range(n - 1):
    label = 0
    print(f'*** Url {i}'.format(dataframeUrl.iloc[i]['url']))
    if dataframeUrl.iloc[i]['result'] == 0:
        label = 1
    else:
        label = -1
    aux = feature_extraction40.website(dataframeUrl.iloc[i]['url'], label)
    aux.getFeatures()
    print(aux.features)
    if aux.doubt == 0:
        websiteList.append(aux.features)
    else:
        listDoubt.append(dataframeUrl.iloc[i])

dtFinish = pd.DataFrame(websiteList, columns=['haveIp', 'lengthUrl', 'haveAtSymbol', 'sslState', 'domainAge', 'slashDouble', 'anchorUrl', 'prefixSuffix', 'linksInTags', 'clickRight',
'windowsPopUp', 'favicon', 'abnormalURL', 'iframe', 'dnsRegister', 'googleIndex', 'port', 'requestUrl', 'sfh', 'websiteForwarding',
'mouseOver', 'webTraffic', 'shorterService', 'domainRegisterAge', 'httpsToken', 'emailInform', 'pageRank', 'staticInform', 'haveSubdomain', 'linksToPage',
'hasPDS', 'hasSHA1', 'hasYara', 'hasSHA256', 'hasShort', 'hasDateTime',
'hasDomain', 'hasHostname', 'hasIPDst', 'hasIPSrc', 'result'])

dtFinish.to_csv('dataset_part2.csv', index_label='Ord.')

print('Lista de url que dieron problemas')
print(listDoubt)
```

Fig. 15. Código del algoritmo para analizar las 40 características

```
Modelos 30 caracteristicasyphib M cleanDataset.py M Modelos prueba caracteristicasyphib data_1.csv U datasetlegitimephishing-ok.csv U
crea_dataset > datasetlegitimephishing-ok.csv
1 rec_id,url,website,result,created_date
2 1,https://www.mathopenref.com/segment.html,1635698138155948.html,0,10/31/2021 16:35
3 2,https://www.computerhope.com/issues/ch000254.htm,1635699278889766.html,0,10/31/2021 16:53
4 3,https://jobs.emss.org.uk/lcc-aspa,16356510250721.html,0,2/17/2021 18:01
5 4,https://best-mac-tips.com/2014/08/14/changing-your-mac-address-on-ios-iphone-ipad/,163570355325743.html,0,10/31/2021 18:05
6 5,https://singup.live.com/211ca1,1626171244686112.html,0,7/13/2021 15:44
7 6,https://forbusiness.snapchat.com/,1626720628315328.html,0,7/19/2021 18:50
8 7,https://www.artccw.com/diy/mousepad,1635699761652684.html,0,10/31/2021 17:02
9 8,http://www.academickids.com/encyclopedla/index.php/Connectedness,1635704538042534.html,0,10/31/2021 18:22
10 9,https://signalprocessingsociety.org/professional-development/jobs-signal-processing,163570407542444.html,0,10/31/2021 18:27
11 10,https://okpay.livejournal.com/8703.html,1613581349027434.html,0,2/17/2021 22:32
12 11,https://www.gamespot.com/genre/adventure/pc/,1635706054338702.html,0,10/31/2021 18:47
13 12,https://www.screamingfrog.co.uk/pay-per-click/,1635703174277606.html,0,10/31/2021 17:59
14 13,https://www.adelaide-tickets.com/adelaide-tickets/artist/860931,1635707612573897.html,0,10/31/2021 19:13
15 14,https://c.folklink.net/wiki/Category:Gateway_Arch,1635700926348099.html,0,10/31/2021 17:22
16 15,https://systematicreviewsjournal.biomedcentral.com/,161355552947897.html,0,2/17/2021 15:22
17 16,https://probearoundthelobe.com/costs-for-a-weekend-in-alicante-spain/,161351087043553.html,0,2/17/2021 8:34
18 17,https://www.pocketpence.co.uk/motivation-important-4579837.html,1635701813488794.html,0,10/31/2021 17:36
19 18,https://www.ny1.com/image/228799/corpus-christi-a-picture-of-bascule-bridge-at-night-in-corpus,163571190852022.html,0,10/31/2021 20:25
20 19,https://financialit.net/,1623153787831341.html,0,08/06/2021 17:33:00
21 20,https://linkscape.org/gallery/-extensions/,1623144913839212.html,0,08/06/2021 15:05:00
22 21,https://hosting-orange-eg/resent.php,1609089837865107.html,0,12/27/2020 22:53
23 22,https://www.reynigroup.com/,162611472324731.html,0,7/13/2021 0:02
24 23,https://security.berkeley.edu/secure-coding-practice-guidelines,1635710967076669.html,0,10/31/2021 20:09
25 24,http://textfilesplitter.org/,1635702112841136.html,0,10/31/2021 17:41
26 25,https://www.slideserve.com/yardley/input-output-model,1635701527188398.html,0,10/31/2021 17:32
27 26,https://stackoverflow.com/help/reset-password,1626918683178001.html,0,7/22/2021 1:51
28 27,https://www.akalipro.com/softwaredownload,1609103057081006.html,0,12/28/2020 2:34
29 28,https://www.glassdoor.com.au/Job/e-banking-specialist-jobs-SRCH_K00_20_IP2.htm,1613555694503998.html,0,2/17/2021 15:24
30 29,https://www.washos.com/login,1613583893042525.html,0,2/17/2021 23:14
31 30,http://www.discovertz.com/,1613565519333911.html,0,2/17/2021 18:08
32 31,https://www.kik.com/login,1609099958251125.html,0,12/28/2020 1:42
33 32,https://isis.anu.edu.au/,1609112874856506.html,0,12/28/2020 5:17
34 33,https://www.temp-auth.org.ii/,1613528474006623.html,0,2/17/2021 7:51
35 34,https://staffmp.com/jobs,1635702682404449.html,0,10/31/2021 17:51
36 35,https://accounts.google.com/ServiceLogin?passive=120960080sId=1kcontinue=https://plus.google.com/2281linkedIn&followup=https://plus.google.com/2281linkedIn,1635709775771
37 36,https://www.tomaddead.net/link-to-the-iv-doc-an-innovative-e-commerce-home-healthcare-company/,1613518953095924.html,0,2/17/2021 5:12
38 37,https://www.allabouttabletennis.com/rules-of-table-tennis.html,1635699538379641.html,0,10/31/2021 16:58
39 38,https://www.youtube.com/user/bodyguardshoppers,1635711762729733.html,0,10/31/2021 20:22
40 39,https://help.fibbit.com/articles/en-US/help/articles/18973.htm,1609184424858917.html,0,12/28/2020 2:57
41 40,https://myhostosandprojects.blogspot.com/2014/11/wake-on-lan-linux.html,163570745914582.html,0,10/31/2021 19:10
42 41,https://www.cedarpoint.com/park-map-and-directions,1623148218444173.html,0,08/06/2021 16:00:00
43 42,https://www.easytechjunkie.com/what-is-computer-memory.htm,1635708073041913.html,0,10/31/2021 19:21
44 43,https://www.universallorlando.com/web/green,1635712690877556.html,0,10/31/2021 20:38
45 44,https://www.macworld.co.uk/feature/best-lightning-headphone-adapters-for-iphone-3646344/,163570527190685.html,0,10/31/2021 18:34
46 45,https://hypestat.com/host/AS2222/FR,1635702736421968.html,0,10/31/2021 17:52
47 46,https://www.airexplorer.net/en/download/,1613545032238781.html,0,2/17/2021 12:27
48 47,https://www.spencersrestaurant.com/,1626121683834194.html,0,7/13/2021 1:58
```

Fig. 15. Depuración del código realizado

Se puede evidenciar la creación del dataset en base a una data que se descargó directamente del Repositorio de Kaggle esta misma contenía todas las URL de phishing y no phishing. Cabe destacar que también se proporciona el código, con el cual se obtuvo el dataset, este mismo contiene un total de 40 características de extracción que se puede obtener de una URL dando como resultado 30 características propias y 10 características con IOC.

Una vez terminado el entrenamiento del dataset, este se guardará automáticamente en un archivo csv en el cual se tendrá un total de 40 características obtenidas por cada URL ingresada proveniente del dataset descargado del repositorio, además poseerá los valores de 1 y -1 respectivamente como se muestra en la figura 16.

Figura 16

Dataset Creado

```

Modelos 30 características.ipynb M cleanDataset.py M Modelos prueba características.ipynb data_1.csv U dataset-phishing-legitime3.csv X
modeloSeleccionado > dataset-phishing-legitime3.csv
1 Ord.,haveIp,lengthInI,haveAtSymbol,sslState,domainAge,slashDouble,anchorUrl,prefixSuffix,linksInTags,clickRight,windowsPopUp,favicon,abnormalURL,
2 0,1,1,1,1,1,1,-1,1,0,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,-1,1,1
3 1,1,1,1,1,1,1,-1,1,0,-1,-1,1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,1,-1,-1,1,-1,1
4 2,1,1,1,1,1,1,-1,1,-1,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,1
5 3,1,-1,1,1,1,1,1,-1,-1,-1,1,-1,1,1,1,-1,0,1,-1,-1,-1,1,1,1,-1,-1,1,1,-1,1
6 4,1,1,1,1,1,1,-1,1,1,-1,1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
7 5,1,1,1,1,1,1,0,1,0,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
8 6,1,1,1,1,1,1,-1,1,0,-1,1,1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
9 7,1,0,1,-1,1,1,0,1,0,-1,-1,1,-1,1,1,1,-1,-1,0,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,1
10 8,1,-1,1,1,1,1,-1,1,0,-1,-1,-1,-1,1,1,1,-1,0,0,-1,-1,-1,1,1,1,-1,-1,1,1,-1,1
11 9,1,1,1,1,1,1,0,1,0,-1,-1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,1
12 10,1,1,1,1,1,-1,1,-1,-1,-1,-1,1,1,1,-1,-1,0,-1,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
13 11,1,1,1,1,1,1,1,1,-1,-1,-1,-1,1,1,1,-1,-1,0,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,1
14 12,1,0,1,1,-1,1,0,1,1,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,-1,1,-1,-1,1,-1,-1,1
15 13,1,0,1,1,1,1,-1,1,0,-1,-1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
16 14,1,1,1,1,1,1,-1,1,-1,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,1,1,0,-1,1
17 15,1,0,1,1,1,1,0,1,0,-1,-1,-1,-1,1,1,1,-1,0,1,-1,-1,-1,1,1,1,-1,-1,1,1,-1,1
18 16,1,0,1,1,1,1,0,1,-1,-1,-1,1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,1
19 17,1,-1,1,1,1,1,-1,1,1,-1,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,-1,1,1,-1,-1,1,0,-1,1
20 18,1,1,1,1,1,-1,1,0,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,1,-1,1
21 19,1,1,1,1,1,-1,1,0,-1,-1,1,-1,1,1,1,-1,-1,0,-1,-1,-1,1,1,1,-1,1,1,1,-1,1
22 20,1,1,1,1,-1,1,-1,1,0,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,-1,1,-1,1,-1,-1,1,-1,1
23 21,1,1,1,1,1,0,1,-1,-1,-1,1,-1,1,1,1,-1,-1,1,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
24 22,1,0,1,1,-1,1,-1,1,0,-1,-1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,1,-1,1,-1,-1,1,0,-1,1
25 23,1,1,1,-1,1,1,-1,1,1,-1,1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,1,-1,1,1,-1,1
26 24,1,1,1,1,1,-1,1,0,-1,-1,1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,1,-1,1,1,0,-1,1
27 25,1,1,1,1,1,-1,1,1,-1,-1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,1,1,1,-1,1
28 26,1,1,1,1,1,0,1,1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
29 27,1,-1,1,1,-1,1,-1,1,1,-1,-1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,1,-1,1,-1,-1,1,-1,-1,1
30 28,1,1,1,1,1,0,1,0,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
31 29,1,1,1,-1,1,1,1,1,-1,-1,-1,1,-1,1,1,1,-1,-1,1,-1,-1,-1,1,1,1,-1,1,0,-1,1
32 30,1,1,1,1,1,-1,1,-1,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,-1,1,1,0,1,1
33 31,1,1,1,1,-1,1,1,1,-1,-1,-1,1,-1,1,1,1,-1,1,0,0,-1,-1,1,-1,1,-1,-1,1,-1,1,1
34 32,1,1,1,1,-1,1,0,-1,-1,-1,-1,-1,1,1,1,-1,-1,1,-1,-1,-1,1,-1,1,-1,-1,1,-1,1,1
35 33,1,1,1,1,1,-1,1,0,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,-1,-1,1,1,-1,1,1
36 34,1,-1,1,1,1,-1,0,1,1,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,-1,1,-1,1,1
37 35,1,-1,1,1,1,1,1,1,0,-1,-1,1,-1,1,1,1,-1,-1,1,-1,-1,-1,1,1,-1,1,1,0,-1,1,1
38 36,1,0,1,1,1,1,1,0,-1,-1,1,-1,1,1,1,-1,-1,1,-1,-1,-1,1,1,1,1,-1,1,1,-1,1,1
39 37,1,1,1,1,1,1,0,1,-1,-1,1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,1,1,-1,1,0,1,1
40 38,1,0,1,1,1,-1,1,1,1,-1,-1,1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,1,1,1,-1,1,-1,-1,1
41 39,1,0,1,-1,-1,1,0,1,0,-1,-1,-1,-1,1,-1,1,-1,1,1,-1,-1,-1,1,-1,-1,1,-1,-1,1,1
42 40,1,1,1,1,1,1,1,1,1,-1,-1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,-1,1,1,-1,1,1,0,-1,1
43 41,1,0,1,1,1,1,1,0,-1,-1,1,-1,1,1,1,-1,-1,0,-1,-1,-1,1,1,1,1,-1,-1,1,-1,-1,1
44 42,1,1,1,1,1,1,1,1,0,-1,-1,1,-1,1,1,1,-1,1,1,-1,-1,-1,1,1,1,-1,1,1,0,1,1
45 43,1,-1,1,1,1,1,1,-1,1,1,-1,-1,1,-1,1,1,1,-1,0,1,0,-1,-1,1,1,1,1,-1,-1,-1,1
46 44,1,1,1,1,1,1,1,1,1,-1,-1,-1,-1,1,1,1,-1,1,0,-1,-1,-1,-1,1,1,-1,-1,1,1,-1,1
47 45,1,1,1,1,1,1,1,1,1,-1,-1,-1,1,-1,1,1,1,-1,-1,1,-1,-1,-1,1,1,1,-1,-1,1,0,-1,1
48 46,1,1,1,1,1,1,1,1,1,-1,-1,-1,1,-1,1,1,1,-1,1,0,-1,-1,-1,-1,1,1,1,-1,1,1,0,-1,1

```

Se verifica correctamente con cada uno de los 6 modelos seleccionados cuál es el mejor en la detección y prevención de sitios web con phishing.

Usando como fuente principal la generación del dataset entrenado con anterioridad. Se realizó las pruebas del dataset creado con los respectivos modelos y se registró cada uno de los registros respectivamente además se puede evidenciar en la figura 17

Figura 17

Pruebas con Respectivos Modelos/Algoritmos de Machine Learning

```

df = pd.read_csv("dataset.csv",index_col=0)
#df = sklearn.utils.shuffle(df)
X = df.drop("Result",axis=1).values
X = preprocessing.scale(X)
y = df['Result'].values
df.head()
}

def mean_score(scoring):
    return {i:j.mean() for i,j in scoring.items()}
}

scoring = {'accuracy': 'accuracy',
           'recall': 'recall',
           'precision': 'precision',
           'f1': 'f1'}
fold_count=10
}

#Random Forest
from sklearn.ensemble import RandomForestClassifier
rforest_clf = RandomForestClassifier()
cross_val_scores = cross_validate(rforest_clf, X, y, cv=10, scoring = scoring)
rforest_clf_score = mean_score(cross_val_scores)
print(rforest_clf_score)
}

{'fit_time': 0.5254746913909912, 'score_time': 0.02170724868774414, 'test_accuracy': 0.9722253770057195, 'test_recall': 0.9806696758526027, 't

#Multi-layer Perceptron classifier
from sklearn.neural_network import MLPClassifier
neural_clf=MLPClassifier(hidden_layer_sizes=(33,),max_iter=500)
cross_val_scores = cross_validate(neural_clf, X, y, cv=fold_count, scoring=scoring)
neural_clf_score = mean_score(cross_val_scores)
print(neural_clf_score)
}

```

Luego de la obtención respectiva del mejor modelo se realizó la creación y el contrato de un servidor para subir el aplicativo. El servidor en donde se usará el respectivo modelo será PythonAnywhere, en este se subirán varios archivos como el de extracción de características, el app.py y el modelo ya entrenado. Se puede evidenciar en la tabla 18

Figura 18

Servidor PythonAnywhere

The screenshot shows the PythonAnywhere dashboard for user 'george246'. The 'Files' section is active, showing a list of files:

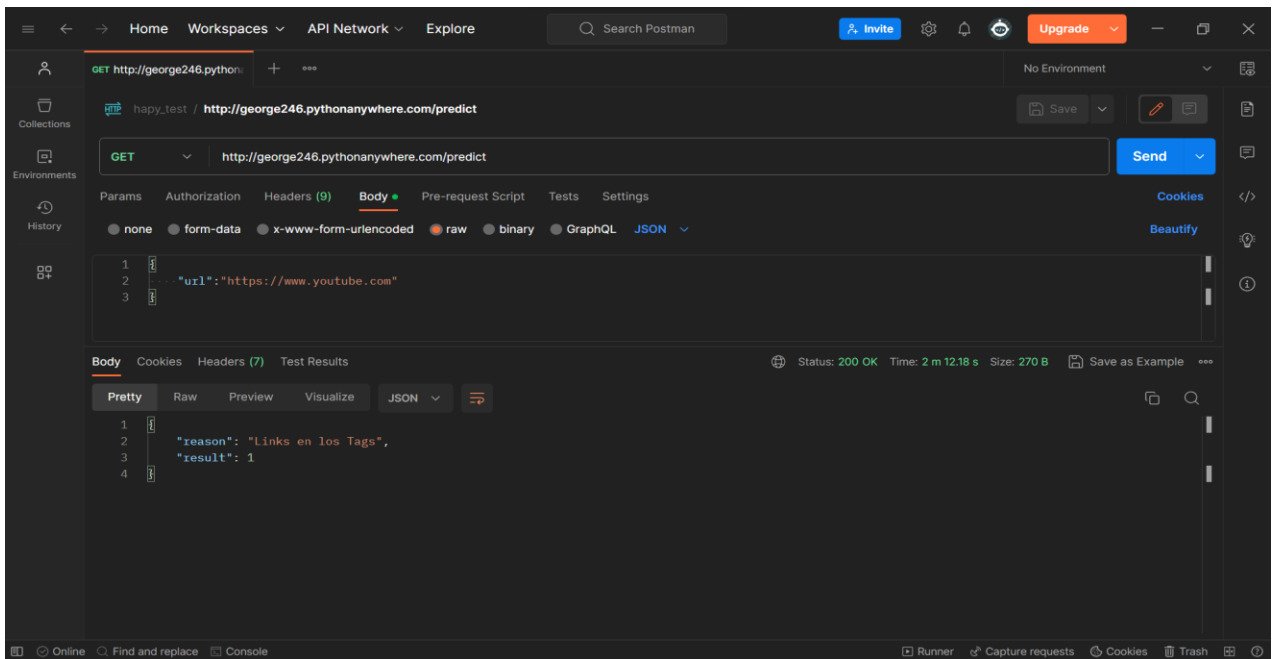
File Name	Download	Trash	Created	Size
app.py			2023-06-26 17:45	3.1 KB
featureExtraction.py			2023-06-08 18:07	19.2 KB
randomForest-model.sav			2023-06-08 18:32	25.2 MB

Below the file list is an 'Upload a file' button with a note: '100MiB maximum size'.

Una vez incorporado los respectivos archivos en el servidor se realizó pruebas usando la herramienta Postman para el uso de peticiones y de esta manera verificar si se logró obtener la respuesta que se esperaba una vez ingresada la URL, cabe mencionar que depende la petición, para el retorno de una respuesta mostrando el valor de 1 o -1 verificando si la URL ingresada, posee o no phishing esto se muestra en la figura 19

Figura 19

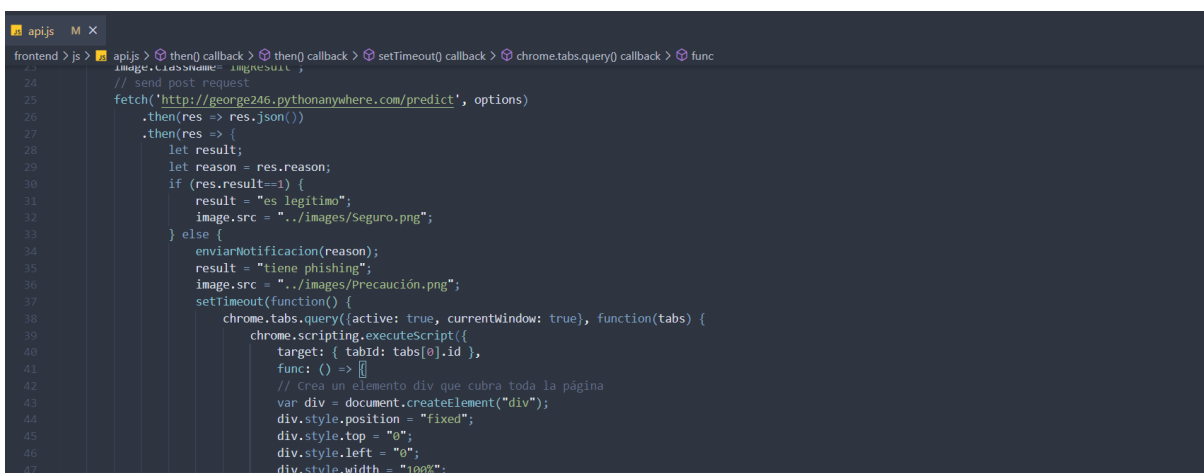
Utilización Postman para verificación respectiva de URL e identificar si tiene Phishing o no



Después de la respectiva verificación con respecto a las peticiones realizadas en el backend se procede a conectar y realizar la parte del frontend incorporando todos los puntos mencionados en la historia de usuario como, detección basada en firmas y en anomalías. Además de incorporación de la extensión de manera gráfica para una mejor visualización de los puntos mencionados, así como la extensión esto se evidencia en una pequeña parte de código mostrada en la figura 20

Figura 20

Frontend conexión con el Backend



Finalmente se muestra la interfaz de la extensión colocada en Google Chrome, además de los puntos mencionados, como el aviso si se tiene existencia de Phishing,

notificación de alerta, además de la respectiva razón del bloqueo de la página mostrando una imagen que menciona página bloquea y cierre respectivo de la misma.

Figura 21

Interfaz Gráfica de la Extensión en Google Chrome

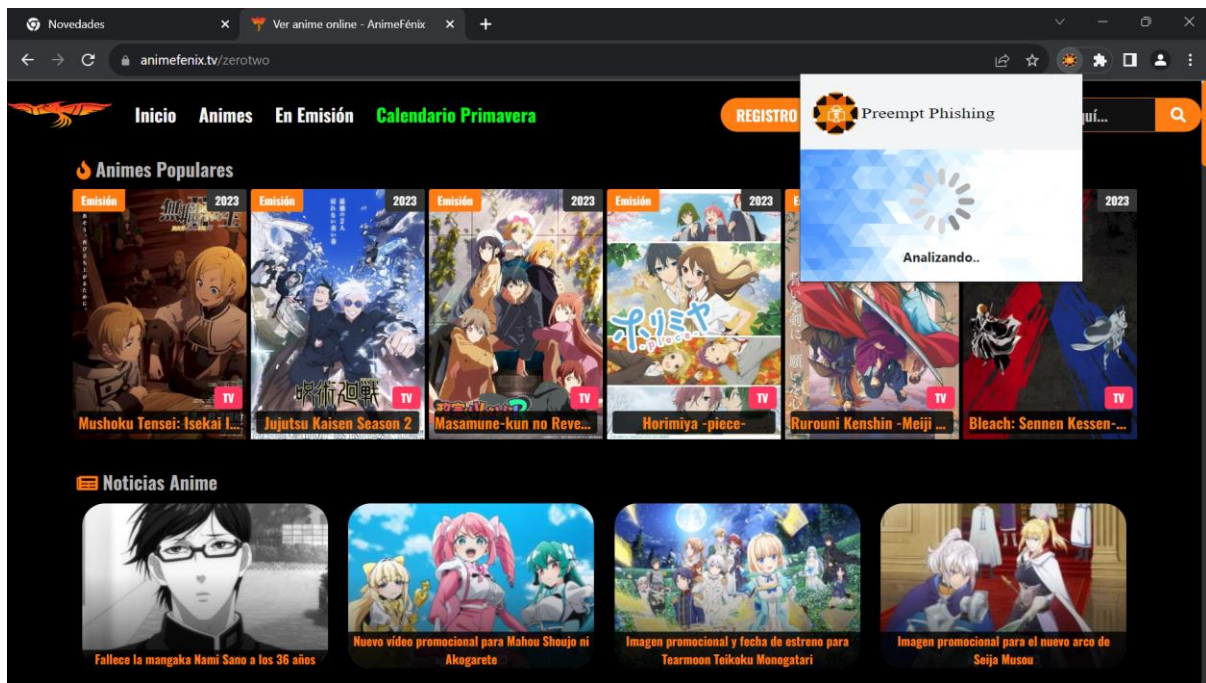


Fig. 21 a Sitio Web donde se utilizó la extensión

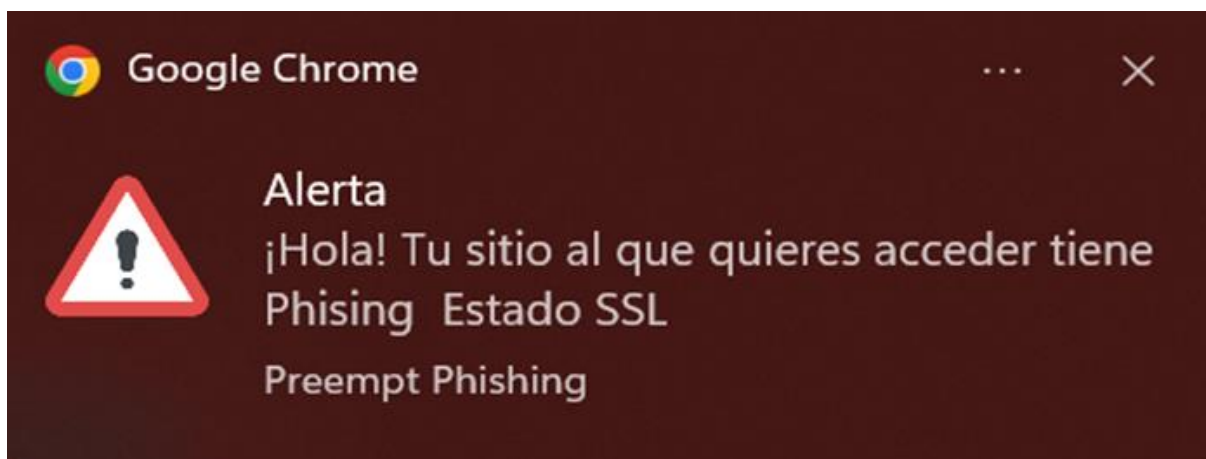


Fig. 21 b. notificación de la extensión al momento de detectar que el sitio web tiene Phishing



Fig. 21 c. Bloqueo de página al detectar phishing en el sitio web.

Resumen del desarrollo del sistema de prevención de sitios web con phishing.

Sprint 01: Después de realizar pruebas con los 6 modelos y algoritmos de Machine Learning mencionado en la tabla 3, se eligió Random Forest como el modelo más adecuado para hacer las predicciones de sitios web con phishing o legítimos. Este Sprint se basó en la búsqueda de una alta y aceptable precisión en la detección de sitios web con phishing.

Sprint 02: Se seleccionaron 40 características que son extraídas a partir de una URL. Además de ser probadas en diferentes escenarios y combinados para encontrar mejores valores del accuracy y garantizar una mejor precisión en la predicción de sitios web con phishing. Además, de la creación de un dataset con las características seleccionadas a partir de otro que contenía sólo las URLs de sitios web legítimos y con phishing.

Sprint 03: El modelo de Machine Learning elegido fue entrenado utilizando el conjunto de datos creado en el Sprint anterior. Después, se desarrolló una API Rest y se implementó en un servidor para su disponibilidad.

Sprint 04: La extensión de Google Chrome fue creada empleando las tecnologías de HTML, CSS y JavaScript. Y se implementó métodos de prevención para los ataques phishing tales como: notificación, cierre. Después, la razón por la que el sitio web tiene

phishing y finalmente, las pruebas en el código junto con correcciones de errores en el sistema.

Capítulo IV

Validación del Sistema

Se lleva a cabo las pruebas necesarias para validar la funcionalidad de la extensión de Google Chrome “Preempt Phishing”. Para este propósito, se utilizó dos herramientas la primera MaxPhisher, la cual está disponible en el sistema Operativo Kali Linux y la segunda Zphisher la cual también dispone del mismo sistema Operativo, sirven para generar sitios web con Phishing a través de una terminal, además, estas herramientas recopilan datos ingresados por el usuario, y finalmente muestran los datos ingresados y su dirección IP. En la figura 22 y en la figura 23, se pueden demostrar los sitios web con Phishing que las aplicaciones pueden generar.

Figura 22

Ataques disponibles MaxPhisher

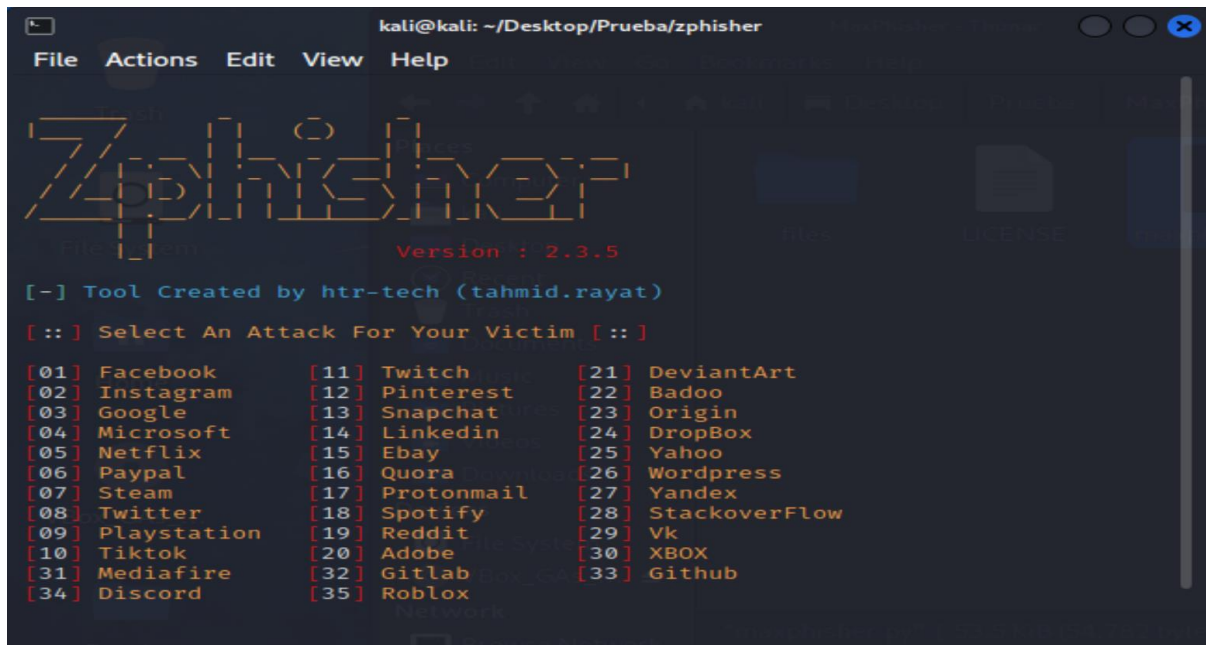
```

MaxPhisher [v1.1] [By KasRoudra]

[01] Facebook Traditional      [27] Reddit                    [53] Gitlab
[02] Facebook Voting          [28] Adobe                      [54] Github
[03] Facebook Security        [29] DevianArt                   [55] Apple
[04] Messenger                 [30] Badoo                       [56] iCloud
[05] Instagram Traditional     [31] Clash Of Clans             [57] Vimeo
[06] Insta Auto Followers      [32] Ajio                       [58] Myspace
[07] Insta 1000 Followers      [33] JioRouter                   [59] Venmo
[08] Insta Blue Verify         [34] FreeFire                   [60] Cryptocurrency
[09] Gmail Old                  [35] Pubg                       [61] SnapChat2
[10] Gmail New                  [36] Telegram                    [62] Verizon
[11] Gmail Poll                 [37] Youtube                     [63] Wi-Fi
[12] Microsoft                  [38] Airtel                      [64] Discord
[13] Netflix                    [39] SocialClub                 [65] Roblox
[14] Paypal                      [40] Ola                         [66] UberEats
[15] Steam                       [41] Outlook                     [67] Zomato
[16] Twitter                     [42] Amazon                      [68] WhatsApp
[17] PlayStation                 [43] Origin                      [69] PhonePay
[18] TikTok                      [44] DropBox                    [70] MobikWik
[19] Twitch                      [45] Yahoo                       [71] FlipCart
[20] Pinterest                   [46] WordPress                   [72] Teachable
  
```

Figura 23

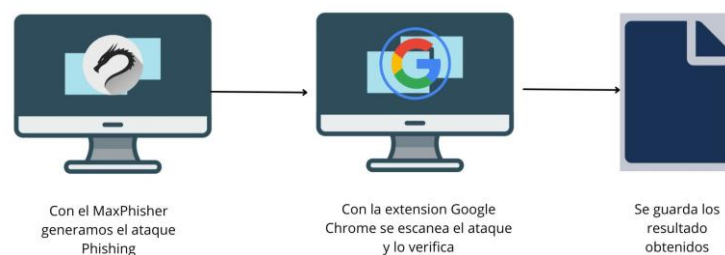
Ataques disponibles Zphisher



Para estas pruebas, se empleó dos computadoras diferentes. En el primer ordenador, se instaló la máquina virtual Virtual Box, con el sistema operativo Kali Linux, la cual fue utilizada para enviar los ataques de phishing. Por otro lado, en el segundo computador, se instaló la extensión de Google Chrome denominada “Preempt Phishing”. Una vez que el ataque phishing se envió desde el primer computador, se accede al sitio web desde el otro para poder inspeccionar y guardar los resultados obtenidos. Esto se procedió a realizar con cada herramienta que se mencionó con anterioridad repitiendo el proceso. La figura 24 muestra de manera visual el proceso de ejecución de estas pruebas. Descritas previamente.

Figura 24

Proceso de pruebas



Definición y evaluación de métricas utilizadas

Aplicación de las métricas de evaluación

Para adaptar las métricas de evaluación en primer lugar se realizó las pruebas de la extensión Preempt Phishing con el primer modelo de Machine Learning entrenado, utilizando las herramientas Zphisher y MaxPhiser, de los cuales se obtuvieron los resultados que se muestran en la tabla 23. La tabla está compuesta por atributos, nombres de secciones específicas del sitio web, nombre del sitio web, resultado esperado y predicción del modelo seleccionado. En esta tabla se registraron las pruebas del sistema de detección de ataques Phishing, por lo cual en ambas herramientas se seleccionaron las características similares dando como resultado 35 sitios web con Phishing generados principalmente por la herramienta Zphisher y 35 sitios web que pertenecen a la misma selección del sitio web pero Legítimos.

Los sitios web que se seleccione se muestran en la figura 23 y son en total 35, estos están disponibles en Zphisher, tal como se muestra en la columna "SITIO WEB" en la tabla 23, además de los sitios web que se muestran en la figura 22 y en esta son un total de 72, los cuales se muestran de en la columna antes mencionada pero en la tabla 26, para una mejor comprensión de los sitios web ingresados se categorizo de acuerdo a la plataforma o subsistemas de que controla la funcionalidad como se muestra en la tabla 22.

Tabla 22

Categorización de Sitios Web

Redes Sociales	Facebook, Instagram, Facebook Traditional, Facebook Voting, Facebook Security, Messenger, Instagram Tradicional, Insta Auto Followers, Insta 1000 Followers, Insta Blue Verify, Twitter, Tik Tok, Pinterest, Snapchat, Snapchat2, LinkedIn, Quora, DeviantArt, Badoo, Vk, Telegram, WhatsApp
Plataforma SSO (Single	Google, Adobe, Microsoft, Yahoo!, Airtel, El Acuerdo de Nivel

Sign On)	Operacional (OLA),
Plataforma de streaming por suscripción	Netflix
Plataformas de pagos on line	Paypal, Amazon, Venmo, Cryptocurrency, UberEats, PhonePay, Mobikwik, Flipcart
Plataformas de distribución digital de videojuego	Steam, PlayStation, Origin, XBOX, Apple
Plataforma de streaming de video	Twitch, Ajio GameZone, YouTube, Vimeo
Portal web para vender y/o subastar on line	eBay
Plataforma de Internet	JioRouter, Wi-Fi
Plataforma de desarrollo	WordPress, Teachable
Juegos	Clash of Clans, Free Fire, Pubg, Roblox
Correo electrónico	Prontomail, Yandex, Gmail Old, Gmail New, Gmail Poll, Outlook
Plataforma de streaming de música:	Spotify
Plataforma social	Reddit, Discord, SocialClub, Myspace, Zomato
Servicio de alojamiento de archivos	DropBox, MediaFire, iCloud
Blog	Stackoverflow
Gestor de versiones	Github, Gitlab

Nota. Se presenta los sitios categorizados.

En la mayor parte de los Sitios se tomó en cuenta la sección Login Page debido a que es la primera interfaz que el usuario visualiza e ingresa sus datos personales y privados dentro de un sistema web en general. El objetivo de los Phisher es engañar a sus víctimas intentan obtener la información confidencial de los usuarios, como nombres de usuario contraseñas, y detalles de la tarjeta de crédito (Wu et al.,2020) todo esto con fines ilegales, sin embargo, para completar los 74 sitios distribuidos entre Zphisher y MaxPhiser, se seleccionaron más de una sección de varios sitios web, los cuales se basaron en secciones que requieren ingreso por parte del usuario, los cuales tanto Zphisher y MaxPhisher pueden generar.

Para entender cómo se procedieron con la ejecución de las pruebas del sistema en esta sección se explica brevemente el procedimiento para el 1 Sitio Web que es Facebook, para ello se tomó una sección, en específico para ser probados, las cuales podrían ser obtenidos mediante las herramientas Zphisher y MaxPhiser. Se obtuvieron principalmente las secciones de Sitios Web con phishing, a través de Zphisher: Tradicional Login Page (Página de sesión de Facebook). Después se buscó la misma sección del mismo Sitio Web legítimo y de todo se registró la URL. Luego se probó cada URL con la extensión Preempt Phishing desarrollada con el primer modelo de Machine Learning implementado en el capítulo 3: Implementación del Sistema. Se probó de esta manera todos los demás Sitios Web con sus respectivas secciones. Finalmente, se puede observar, en la tabla 23 y 26, que el modelo únicamente detecta Sitios Web con Phishing, en cuanto a los Legítimos este sistema los considera como Phishing.

Tabla 23

Resultados pruebas de Preempt Phishing junto con ZPhisher con modelo entrenado de Machine Learning

Sitio Web	Ord.	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultado esperado	Predicción	Resultado esperado	Predicción
Facebook	1	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Instagram	2	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Google	3	Tradicional Login page	Phishing	Legítimo	Legítimo	Legítimo
Microsoft	4	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Netflix	5	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
PayPal	6	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Steam	7	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Twitter	8	Tradicional Login page	Phishing	Legítimo	Legítimo	Legítimo
PlayStation	9	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
TikTok	10	Tradicional Login page	Phishing	Legítimo	Legítimo	Legítimo
Mediafire	11	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Discord	12	Tradicional	Phishing	Phishing	Legítimo	Legítimo

Sitio Web	Ord.	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultado esperado	Predicción	Resultado esperado	Predicción
		Login page				
Twitch	13	Tradicional Login page	Phishing	Phishing	Legitimo	Legitimo
Pinterest	14	Tradicional Login page	Phishing	Legitimo	Legitimo	Phishing
Snapchat	15	Tradicional Login page	Phishing	Phishing	Legitimo	Legitimo
Linkedin	16	Tradicional Login page	Phishing	Phishing	Legitimo	Legitimo
Ebay	17	Tradicional Login page	Phishing	Legitimo	Legitimo	Legitimo
Quora	18	Tradicional Login page	Phishing	Legitimo	Legitimo	Legitimo
Protonmail	19	Tradicional Login page	Phishing	Legitimo	Legitimo	Legitimo
Spotify	20	Tradicional Login page	Phishing	Phishing	Legitimo	Legitimo
Reddit	21	Tradicional Login page	Phishing	Phishing	Legitimo	Legitimo
Adobe	22	Tradicional Login page	Phishing	Phishing	Legitimo	Phishing
Gitlab	23	Tradicional Login page	Phishing	Phishing	Legitimo	Legitimo

Sitio Web	Ord.	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultado esperado	Predicción	Resultado esperado	Predicción
Roblox	24	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
DeviantArt	25	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Badoo	26	Tradicional Login page	Phishing	Phishing	Legítimo	Phishing
Origin	27	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
DropBox	28	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Yahoo	29	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
WordPress	30	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Yandex	31	Tradicional Login page	Phishing	Phishing	Legítimo	Phishing
Stackoverflow	32	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
VK	33	Tradicional Login page	Phishing	Phishing	Legítimo	Phishing
XBOX	34	Tradicional Login page	Phishing	Phishing	Legítimo	Legítimo
Github	35	Tradicional	Phishing	Phishing	Legítimo	Legítimo

Sitio Web	Ord.	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultado esperado	Predicción	Resultado esperado	Predicción
Login page						
Sitios Web Bien Clasificados			29		30	
Sitios Web Mal Clasificados			6		5	

Nota. Se presenta los resultados de los sitios web de Zphisher utilizados para la verificación del sistema.

Con los resultados presentados en la tabla 23, se procede realizar la matriz de confusión correspondiente como se puede ver en la tabla 24

Tabla 24

Matriz de Confusión del modelo ML usando Zphisher

	POSITIVOS	NEGATIVOS
POSITIVOS	29 (VP)	30 (FP)
NEGATIVOS	6 (FN)	5 (VN)

Ahora, las fórmulas para calcular las métricas de Accuracy, Precision, Recall y F1, que están referenciadas en la tabla 4, los resultados que se obtuvieron se muestran en la tabla 25. En esta tabla se puede observar que la detección de sitios web con phishing, alcanzando un nivel de precisión del 84% en la métrica Accuracy. Esto significa que más del 75% de los sitios web, son correctamente identificados en relación con el conjunto total de datos. Con respecto a la métrica Precision, se logra un 82% de exactitud en la detección precisa de sitios web con phishing. Por otra parte, la métrica Recall llega al 85% que se detectan adecuadamente la mayoría de los sitios web con phishing en comparación con los sitios web que realmente son de phishing. En cuanto a la métrica F1, el resultado es del

83%, que demuestra que el modelo tiene un mejor rendimiento en la predicción de la clase positiva.

Tabla 25

Métricas de evaluación calculadas

Métricas de Evaluación	Resultado
Accuracy	0,84 (84%)
Precision	0,82 (82%)
Recall	0.85 (85%)
F1	0.83 (83%)

Nota. Se presenta los resultados de las métricas de evaluación utilizadas.

Tabla 26

Resultados pruebas de Preempt Phishing junto con MaxPhisher con modelo entrenado de Machine Learning

Sitio Web	Orden	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultado	Predicción	Resultado	Predicción
			o	n	o	n
			esperado	esperado	esperado	esperado
Facebook	1	Login page	Phishing	Phishing	Legitimo	Legitimo
Messenger	2	Login page	Phishing	Phishing	Legitimo	Legitimo
Instagram	3	Login page	Phishing	Phishing	Legitimo	Legitimo
Gmail	4	Login page	Phishing	Phishing	Legitimo	Legitimo
Microsoft	5	Login	Phishing	Phishing	Legitimo	Legitimo

Sitio Web	Orden	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultad o esperado	Predicció n	Resultad o esperado	Predicció n
					page	
Netflix	6	Login page	Phishing	Phishing	Legitimo	Legitimo
PayPal	7	Login page	Phishing	Phishing	Legitimo	Legitimo
Steam	8	Login page	Phishing	Phishing	Legitimo	Legitimo
Twitter	9	Login page	Phishing	Phishing	Legitimo	Legitimo
PlayStation	10	Login page	Phishing	Phishing	Legitimo	Legitimo
TikTok	11	Login page	Phishing	Phishing	Legitimo	Legitimo
Twitch	12	Login page	Phishing	Phishing	Legitimo	Legitimo
Pinterest	13	Login page	Phishing	Phishing	Legitimo	Phishing
Reddit	14	Login page	Phishing	Phishing	Legitimo	Legitimo
Adobe	15	Login page	Phishing	Phishing	Legitimo	Phishing
DevianArt	16	Login	Phishing	Phishing	Legitimo	Legitimo

Sitio Web	Orden	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultado	Predicción	Resultado	Predicción
			o esperado	n	o esperado	n
		page				
Badoo	17	Login page	Phishing	Phishing	Legitimo	Phishing
Clash of Clans	18	Login page	Phishing	Legitimo	Legitimo	Legitimo
Ajio	19		Phishing	Phishing	Legitimo	Legitimo
JioRouter	20	Login page	Phishing	Phishing	Legitimo	Legitimo
FreeFire	21	Login page	Phishing	Phishing	Legitimo	Phishing
Pugb	22	Login page	Phishing	Phishing	Legitimo	Legitimo
Telegram	23	Login page	Phishing	Phishing	Legitimo	Phishing
YouTube	24	Login page	Phishing	Phishing	Legitimo	Legitimo
Airtel	25	Login page	Phishing	Phishing	Legitimo	Legitimo
SocialClub	26	Login page	Phishing	Phishing	Legitimo	Legitimo
Ola	27	Login page	Phishing	Phishing	Legitimo	Phishing

Sitio Web	Orden	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultado	Predicción	Resultado	Predicción
			o esperado	n	o esperado	n
Outlook	28	Login page	Phishing	Phishing	Legitimo	Legitimo
Amazon	29	Login page	Phishing	Phishing	Legitimo	Legitimo
Origin	30	Login page	Phishing	Phishing	Legitimo	Legitimo
DropBox	31	Login page	Phishing	Legitimo	Legitimo	Legitimo
Yahoo	32	Login page	Phishing	Legitimo	Legitimo	Legitimo
WordPress	33	Login page	Phishing	Phishing	Legitimo	Legitimo
Gitlab	34	Login page	Phishing	Phishing	Legitimo	Legitimo
Github	35	Login page	Phishing	Phishing	Legitimo	Legitimo
Apple	36	Login page	Phishing	Phishing	Legitimo	Legitimo
iCloud	37	Login page	Phishing	Phishing	Legitimo	Legitimo
Vimeo	38	Login page	Phishing	Phishing	Legitimo	Phishing

Sitio Web	Orden	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultad o esperado	Predicció n	Resultad o esperado	Predicció n
			Myspace	39	Login page	Phishing
Venmo	40	Login page	Phishing	Phishing	Legitimo	Legitimo
Cryptocurre ncy	41	Login page	Phishing	Phishing	Legitimo	Phishing
SnapChat2	42	Login page	Phishing	Phishing	Legitimo	Phishing
Verizon	43	Login page	Phishing	Phishing	Legitimo	Legitimo
Wi-Fi	44	Login page	Phishing	Phishing	Legitimo	Legitimo
Discord	45	Login page	Phishing	Phishing	Legitimo	Legitimo
Roblox	46	Login page	Phishing	Legitimo	Legitimo	Legitimo
UberEats	47	Login page	Phishing	Phishing	Legitimo	Legitimo
Zomato	48	Login page	Phishing	Phishing	Legitimo	Phishing
WhatsApp	49	Login page	Phishing	Phishing	Legitimo	Legitimo

Sitio Web	Orden	Sección del sitio Web	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			Resultado	Predicción	Resultado	Predicción
			o esperado	n	o	n
FlipKart	50	Login page	Phishing	Phishing	Legitimo	Legitimo
Teachable	51	Login page	Phishing	Phishing	Legitimo	Legitimo
Sitios Web Bien Clasificados			47		41	
Sitios Web Mal Clasificados			4		10	

Nota. Se presenta los resultados de los sitios web de MaxPhisher utilizados para la verificación del sistema.

Con los resultados presentados en la tabla 26, se procede realizar la matriz de confusión correspondiente como se puede ver en la tabla 27

Tabla 27

Matriz de Confusión del modelo ML usando MaxPhiser

	POSITIVOS	NEGATIVOS
POSITIVOS	47 (VP)	41 (FP)
NEGATIVOS	4 (FN)	10 (VN)

A continuación, se aplican las fórmulas para calcular las métricas Accuracy, Precision, Recall y F1, referenciadas en la tabla 3, y los resultados se muestran en la tabla 28, donde se puede observar que la tasa de detección de sitios web con phishing es notablemente alta, llegando a un 86% de precisión en la métrica Accuracy. Significa que más de los tres cuartos de sitios web, tanto legítimos como phishing, son correctamente identificados en relación con el total de datos de entrenamiento. En términos de la métrica

Precisión, la detección precisa de sitios web con phishing es del 92%. Por otro lado, la métrica Recall alcanza el 82%, esto significa que se considera la mayoría de los sitios con phishing identificados correctamente en comparación con los sitios web que realmente son phishing. Para la métrica F1 el resultado es del 86% que demuestra que predice mejor la clase positiva.

Tabla 28

Métricas de evaluación calculadas

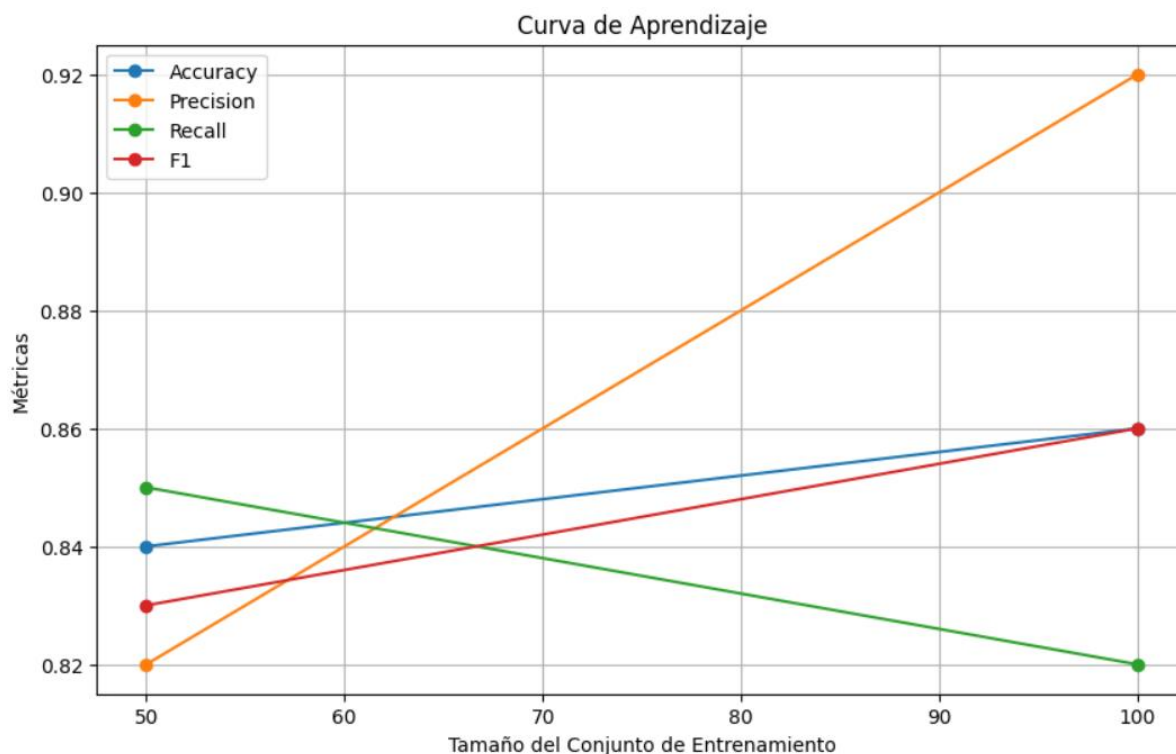
MÉTRICAS DE EVALUACIÓN	RESULTAD O
Accuracy	0,86 (86%)
Precision	0,92 (92%)
Recall	0.82 (82%)
F1	0.86 (86%)

Debido a los resultados presentados en las Métricas de evaluación de las tablas 25 y 28 se presenta la figura 25, se realizó un análisis de rendimiento de modelos Machine Learning utilizando diferentes tamaños de conjuntos de entrenamiento. Se evaluaron cuatro métricas clave: Accuracy, Precision, Recall y F1-score. Los tamaños de conjunto de entrenamiento fueron de 50 a 100 ejemplos respectivamente.

Los resultados indican que a medida que se aumenta el tamaño del conjunto de entrenamiento de 50 a 100 ejemplos, las métricas de rendimiento también mejoran. En términos de Accuracy, se observó un aumento del 84% al 86%, mientras que Precisión aumentó del 82% al 92%. La métrica Recall disminuyó ligeramente del 85% al 82%, pero aún se mantuvo relativamente alta. Similarmente, el F1-score aumentó del 83% al 86%.

Figura 25

Curva de Aprendizaje



Las curvas de aprendizaje son una herramienta de diagnóstico muy utilizada en el aprendizaje automático para algoritmos que aprenden de un conjunto de datos de entrenamiento de forma incremental (Brownlee, 2020).

Análisis de resultados

Para validar los resultados obtenidos del modelo de Machine Learning implementado, se procedió a realizar 10 pruebas con dos diferentes herramientas tanto Zphisher y MaxPhisher en la primer herramienta se contó con un total de 70 sitios web de prueba cada una, en donde: el campo simulado (ambiente controlado) contenía 35 sitios web generados con la herramienta Zphisher, que se pueden observar en la columna SECCIÓN DEL SITIO WEB de la tabla 23 y el campo real (ambiente no controlado) contenía los mismos 35 sitios web pero legítimos. Luego con la segunda herramienta se contó con un total de 102 sitios web de prueba cada uno, en donde: el campo simulado (ambiente controlado) contenía 51 sitios web generados con la herramienta MaxPhisher, que se pueden observar en la columna SECCIÓN DEL SITIO WEB de la tabla 26 y el campo real (ambiente no controlado) contenía los mismos 51 sitios web pero legítimos. El

motivo de realizar las repeticiones 10 veces de esta prueba en ambas herramientas es para verificar que las predicciones se mantienen estables y/o con pequeñas variaciones identificando si están dentro del rango aceptable y con esto poder validar el modelo de Machine Learning implementado.

Se realizó las pruebas en un campo simulado y/o real con el modelo de Machine Learning además con la respectiva herramienta Zphisher, con los resultados obtenidos se calculó el promedio de cada métrica de evaluación, obteniendo los siguientes resultados: 84% para Accuracy, 82% para Precisión, 85% para Recall y 83% para F1. Cabe destacar que calcular estas métricas de evaluación permiten evaluar con qué precisión el modelo predice positivamente los resultados de los sitios web, valores que determinan el aceptable de la predicción. Por otro lado, los resultados de las métricas de evaluación obtenidas en la etapa de entrenamiento fueron: 84% para Accuracy, 85.72% para la Precisión, 96.30 % para Recall y 90% para F1. Con estos datos se pudo determinar que no existe una gran variación a excepción de las métricas de Recall y F1, demostrando que el modelo de Machine Learning es confiable en la etapa de validación (campo simulado/real) dando como punto de partida un buen valor con respecto al Accuracy y la Precisión.

Luego, se aplicó el mismo procedimiento con la herramienta MaxPhisher la cual usó el mismo modelo de Machine Learning, del cual se calculó el promedio de cada métrica de evaluación, obteniendo los siguientes resultados: 86% de Accuracy, 92% de Precisión, 82% de Recall y 86% de F1. Por otra parte, como se mencionó con anterioridad los resultados de las métricas en la etapa de entrenamiento fueron: 84% para Accuracy, 85.72% para la Precisión, 96.30 % para Recall y 90% para F1. Con estos datos se pudo determinar una variación menos dispereja que con la anterior herramienta antes utilizada, afirmando por segunda vez que el modelo de Machine Learning es confiable. Estos resultados se pueden analizar de mejor manera en la tabla 29

Tabla 29

Comparación de modelo con respecto a Zphisher y MaxPhisher.

Etapa de entrenamiento				Campo simulado/real				
Accura cy	Precisi on	Reca ll	F1		Accurac y	Precisio n	Recall	F1
				Zphisher	84%	82%	85%	83%
84%	85.72%	96.3 %	90%	MaxPhish er	86%	92%	82%	86%

Nota. Se presenta la comparación entre los dos softwares utilizados para la verificación del sistema.

Con estos resultados se concluye que existe una clara diferencia entre los valores conseguidos a través de pruebas en un campo simulado/real y los valores obtenidos en entrenamiento, confirmando lo concluido en el párrafo anterior. Como se observa en la tabla 23 y en la tabla 26, todos los sitios web con Phishing de prueba fueron etiquetados por el modelo como Phishing. Los valores obtenidos de las métricas de evaluación con respecto a MaxPhisher se puede visualizar una mejora considerable con respecto al uso respectivo de la herramienta Zphisher, especialmente en el Accuracy (86%) que se encarga de determinar el porcentaje de los sitios web Legítimos y con Phishing que son clasificadores positivamente con respecto a todo el conjunto de datos de prueba. Esta métrica también se vio reflejada con respecto al valor obtenido en la etapa de entrenamiento. También se vio una mejora en la métrica precisión (92%) la cual es la encargada de determinar el porcentaje de sitios web con Phishing clasificados correctamente. Con la Métrica Recall (82%) se nota que disminuye, esto sucede a que esta métrica se encarga de determinar el porcentaje de sitios web con Phishing clasificados correctamente con respecto a los sitios únicamente etiquetados como Phishing de los datasets de prueba seleccionados, en conclusión, no toma en cuenta los datos etiquetados como Legítimos. Finalmente, con la Métrica F1 (86%) se nota que sube con respecto a la herramienta Zphisher, pero disminuye en comparación con el valor de la etapa de entrenamiento, esto debido a que fusiona las métricas tanto accuracy con recall entonces depende precisamente de ambos valores, lo

que demuestra es las diferencias en el rendimiento de un clasificador demostrando que el algoritmo de clasificación predice de mejor manera la clase positiva. A diferencia de otros estudios, este se probó en un campo simulado/real (a través de Zphisher y MaxPhisher) y se obtuvo en Accuracy el valor más alto de 86% y el más bajo con 84%, valores que se encuentran dentro de los valores encontrados en la literatura 83% (Noor et al., 2019).

De los resultados obtenidos, se puede indicar que el sistema de prevención de intrusos (IPS) para ataques phishing, implementado mediante modelos y/o algoritmos de Machine Learning, Indicadores de Compromiso, a través de una extensión Google Chrome, presenta resultados que están dentro del rango aceptable de predicciones según la revisión de la literatura antes realizada.

Conclusiones

A continuación, se muestran las conclusiones a las que se llegó en el desarrollo del respectivo trabajo de investigación:

- Se realizó el análisis del estado del arte sobre indicadores de compromiso para obtener información suficiente en base a qué aspecto nos podría ayudar en la prevención de intrusos en páginas o sitios web apoyado por motores de búsqueda en este caso el uso del browser Google Chrome.
- El IPS desarrollado para la prevención de ataques Phishing denominado "Preempt Phishing" se entrenó con un dataset de 13,192 URL de sitios web (3,987 sitios web con Phishing y 9,205 sitios web Legítimos). Para ello, se determinaron y extrajeron 30 características con referencia a la URL junto con 10 características IOC (Indicadores de Compromiso). Se utilizó 6 modelos y/o algoritmos de Machine Learning seleccionados en base a una revisión sistemática. Se probó y analizaron los resultados obtenidos, con respecto a cada modelo, mediante el uso de métricas de evaluación y se eligió el algoritmo de Random Forest por presentar los valores más altos con respecto a cada una de las métricas propuestas como Accuracy, Precision, Recall y F1 utilizadas para la detectar Sitios Web con Phishing y posteriormente prevenirlos, lo que permitió cumplir con el segundo objetivo específico: Desarrollar una extensión de Google Chrome utilizando técnicas de Machine Learning para mejorar la prevención y gestión de seguridad en sitios web.
- Se implementó un sistema de prevención de Phishing, a través del desarrollo de una extensión para Google Chrome, empleando técnicas de Machine Learning, con el fin de predecir si un sitio es legítimo o posee phishing.
- La utilización de la metodología Scrum, aportó de una manera exponencial al cumplir los objetivos de este proyecto, ya que usa un desarrollo flexible, para la planificación de tareas y mejoras en el avance de los Sprints.

- Para validar el IPS (Preempt Phishing), se hizo el uso de dos herramientas Zphisher y MaxPhisher para generar sitios web con Phishing de prueba, donde el modelo entrenado e implementado se obtuvo valores de las métricas de evaluación por cada una de las herramientas, demostrando así que los valores más relevantes se obtuvieron usando el MaxPhisher destacando las métricas de Accuracy y Precision como las más altas en torno a la otra herramienta y entorno de entrenamiento.
- La extensión de Google Chrome desarrollada (Preempt Phishing) puede ser colocada en un entorno real además posee un acoplamiento con respecto a navegadores como Opera y Brave.

Recomendaciones

- Al momento de realizar una revisión de la literatura tener en cuenta las palabras clave que se relacionan en base al tema, con el fin de obtener buenos resultados y artículos relevantes para iniciar la investigación.
- Para la creación de un dataset obtener principalmente datos actuales y de sitios confiables, además de comprobar que las URLs están disponibles, un punto principal es que contenga Indicadores de Compromiso los cuales fueron un punto fuerte en la investigación.
- Se uso la metodología ágil Scrum, debido a que nos da la facilidad de organizar tareas, uso de comunicación por parte de los integrantes además nos da un plus en el trabajo en equipo y experiencia.
- Todo proyecto de software debe llevar un lugar donde se pueda tener a la mano los cambios realizados para este mismo propósito se usó un repositorio para el control de versiones en el proyecto, y así mejorar la colaboración de los miembros del equipo.
- Realizar de pruebas unitarias, con la finalidad de comprobar el funcionamiento y evitar inconvenientes en la etapa de despliegue en producción.

Bibliografía

Aguilar, L. J. (2017). Ciberseguridad: La colaboración público-privada en la era de la cuarta revolución industrial (Industria 4.0 versus ciberseguridad 4.0). Cuadernos de estrategia, 185, 19-64.

Alzubi, J. A., Alshammari, R., y Al-Betar, M. A. (2018). Predicting phishing websites using decision tree. In 2018 9th International Conference on Information and Communication Systems (ICICS) (pp. 277-282). IEEE.

Anupam, S., y Kar, A. K. (2021). Phishing website detection using support vector machines and nature-inspired optimization algorithms. Telecommunication Systems, 76(1), 17-32. Scopus. <https://doi.org/10.1007/s11235-020-00739-w>

Babar, M. A., y Ghani, S. (2021). A Literature Survey on Social Engineering Attacks: Phishing Attack. International Journal of Advanced Science and Technology, 30(05), 1668-1677.

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Bishop, C., & Nasrabadi, N. (s/f). *Pattern Recognition and Machine Learning*. Recuperado el 23 de agosto de 2023, de <https://link.springer.com/book/9780387310732>

Brownlee, L (2020). Learning Curves for Machine Learning. Machine Learning Mastery. from <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>

Castillo Veloz, M., y Chuquitarco Veloz, K. (2023). Sistema de detección de intrusos en sitios web, usando modelos y/o algoritmos de Machine Learning: caso práctico Phishing Google Chrome.

Chen, H., Liu, Y., y Wang, C. (2020). Network Traffic Analysis for Indicators of Compromise in Software. IEEE Transactions on Information Forensics and Security, 15(4), 789-802.

Cobb, Charles G., *The project manager's guide to mastering Agile: Principles and practices for an adaptive approach*. USA: John Wiley y Sons, 2015.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional.

Cost of a data breach 2022. (n.d.). Ibm.com. Retrieved May 7, 2023, from <https://www.ibm.com/reports/data-breach>

Crockford, D. (2008). *JavaScript: The Good Parts*. Script: The Good Parts.

Denis, M. S., Jean-Baptiste. (2021). *Bayesian Networks: With Examples in R* (2a ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/9780429347436>

Dobrica, L., y Niemela, E. (2002). A survey on software architecture analysis methods. *IEEE Transactions on software Engineering*, 28(7), 638-653.

Drzewiecki, W. (2017). Thorough statistical comparison of machine learning regression models and their ensembles for sub-pixel imperviousness and imperviousness change mapping. *Geodesy and Cartography*, 66(2), 171-209.

García, L., Martínez, R., López, A., y Sánchez, P. (2021). Event Logging and Monitoring for Indicators of Compromise in Software. *International Journal of Cybersecurity*, 18(2), 321-336.

García, L., Martínez, R., López, A., y Sánchez, P. (2021). Signature-Based Intrusion Detection System for Real-Time Traffic Analysis. *International Journal of Information Security*, 18(3), 321-336.

Hansen, A. (2019). *Google Chrome Extensions*. En A. Hansen (Ed.), *Chrome: Web Browser for Android, iOS, Windows, Linux, and Mac OS X* (pp. 123-142). Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Science & Business Media.

Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genetic Programming and Evolvable Machines*, 19(1), 305–307.

<https://doi.org/10.1007/s10710-017-9314-z>

Karabatak, M and Mustafa. T, "Performance Comparison of Classifiers on Reduced Phishing Website Dataset," in 2019 7th International Symposium on Digital Forensic and Security (ISDFS), 2019, pp. 1-6, doi: 10.1109/ISDFS.2019.8739432.

Kumar, K. y Kumar, P. (2020). Detecting Phishing Websites: A Machine Learning Approach. En N. Meghanathan, D. Nagamalai, y N. Chaki (Eds.), *Advances in Computer and Computational Sciences* (pp. 179-187). Springer International Publishing.

https://doi.org/10.1007/978-981-15-7181-7_15

Kurnia, R., Ferdiana, R., y Wibirama, S. (2018). Software Metrics Classification for Agile Scrum Process: A Literature Review. 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 174-179.

<https://doi.org/10.1109/ISRITI.2018.8864244>

Lauzon, V. (2012). Neural networks and their applications. *Scholarpedia*, 7(12), 1533.

López Cruces, C. (2016). Diseño e implementación de una aplicación web para el análisis centralizado de logs de seguridad [B.S. thesis].

López, F. J. A., Avi, J. R., y Fernández, M. V. A. (2018). Control estricto de matrices de confusión por medio de distribuciones multinomiales. *Geofocus: Revista Internacional de Ciencia y Tecnología de la Información Geográfica*, (21), 6.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. En *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics: Vol. 5.1* (pp. 281–298). University of California Press.

<https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and-probability/Chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992>

Mahesh, B. (2019). Machine Learning Algorithms -A Review.

<https://doi.org/10.21275/ART20203995>

Matthes, E. (2019). Python Crash Course.

Mueller, J., y Massaron, L. (2016). Machine learning for dummies. Hoboken, New Jersey: John Wiley y Sons, Inc.

Meyer, E. A., y Weyl, E. (2017). CSS: The Definitive Guide.

Ndichu, S., Ozawa, S., Misu, T., y Okada, K. (2018). A Machine Learning Approach to Malicious JavaScript Detection using Fixed Length Vector Representation. 2018-July. Scopus. <https://doi.org/10.1109/IJCNN.2018.8489414>

Noor, U., Anwar, Z., Amjad, T., y Choo, K. K. R. (2019). A machine learning based FinTech cyber threat attribution framework using high-level indicators of compromise. Future Generation Computer Systems, 96, 227-242

Petersen, K. (2022). Neural networks and deep learning: A review of current research and applications. Journal of Artificial Intelligence Research, 65, 625-652.

Preuveneers, D., y Joosen, W. (2021). Sharing machine learning models as engagement indicators for threat intelligence. Computers y Security, 106, 102319. <https://doi.org/10.1016/j.cose.2021.102319>

Rajoub, B. (2020). Chapter 3—Supervised and unsupervised learning. En W. Zgallai (Ed.), Biomedical Signal Processing and Artificial Intelligence in Healthcare (pp. 51-89). Academic Press. <https://doi.org/10.1016/B978-0-12-818946-7.00003-2>

Ray, S. (2019). A Quick Review of Machine Learning Algorithms. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 35-39. <https://doi.org/10.1109/COMITCon.2019.8862451>

Rivero, J. M., Rossi, G., Grigera, J., Burella, J., Luna, E. R., y Gordillo, S. (2010). From mockups to user interface models: an extensible model driven approach. In Current Trends in Web Engineering: 10th International Conference on Web Engineering ICWE 2010 Workshops, Vienna, Austria, July 2010, Revised Selected Papers 10 (pp. 13-24). Springer Berlin Heidelberg.

Rubin, Kenneth S., *Essential Scrum: A practical guide to the most popular Agile process*. USA: Addison-Wesley, 2012

Rojas, E. (2018). *Glosario de los seis términos básicos del Machine Learning*, Retrieved

Sahingoz, O. K., Buber, E., Demir, O., y Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 117, 345-357. Scopus. <https://doi.org/10.1016/j.eswa.2018.09.029>

Sameen, M., Han, K., y Hwang, S. O. (2020). PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System. *IEEE Access*, 8, 83425-83443. <https://doi.org/10.1109/ACCESS.2020.2991403>

Smith, J., Johnson, A., y Lee, K. (2019). Deep Neural Network-Based Intrusion Detection System for Real-Time Traffic Analysis. *Journal of Network Security*, 15(2), 112-129.

Smith, J., Johnson, A., y Lee, K. (2022). Indicators of Compromise in Software: A Comprehensive Review. *Journal of Information Security*, 20(3), 112-129.

Snyder, C. (2003). *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann.

Sönmez, Y., Tuncer, T., Gökal, H., y Avci, E. (2018). Phishing web sites feature classification based on extreme learning machines. 2018-January 1-5. Scopus. <https://doi.org/10.1109/ISDFS.2018.8355342>

Statcounter Global Stats—Browser, OS, Search Engine including Mobile Usage Share. (s. f.). StatCounter Global Stats. Recuperado 27 de mayo de 2023, de <https://gs.statcounter.com/>

Schwaber, K., y Sutherland, J. (2017). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*.

Stack Overflow - where developers learn, share, y build careers. (n.d.). Stack Overflow. Retrieved July 8, 2023, from <https://stackoverflow.com/>

Schwaber, K., and Mike Beedle (2002), *Agilè Software Development with Scrum*.

USA

Thompson, D., y Clark, A. (2019). Malware Detection Techniques and Indicators of Compromise in Software. *Journal of Computer Virology and Hacking Techniques*, 12(1), 45-63.

Wieggers, K., y Beatty, J. (2013). *Software requirements*. Pearson Education.

Wu, J., Yuan, Q., Lin, D., You, W., Chen, W., Chen, C., y Zheng, Z. (2020). Who are the phishers? phishing scam detection on ethereum via network embedding. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(2), 1156-1166.

Zuñia Macancela, E. R., Arce Ramírez, Á. A., Romero Berrones, W. J., y Soledispa Baque, C. J. (2019). Análisis de la seguridad de la información en las pymes de la ciudad de Milagro. *Revista Universidad y Sociedad*, 11(4), 487-492.

Anexos