Sistema de detección de intrusos en sitios web, usando modelos y/o algoritmos de Machine Learning: caso práctico Phishing Google Chrome

Castillo Veloz, Mishell Estefanía y Chuquitarco Velasco, Kevin Jair

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. Carrillo Medina, José Luis, Ph.D

03 de febrero del 2023

Latacunga

Reporte de verificación de contenido

CERTIFICADO DE ANÁLISIS
magister

Tesis_Castillo_Chuquitarco-Antiplagio-Compilation_15-02-2023_

< 1% Similitudes

< 1% Texto entre comillas
0% similitudes entre comillas
< 1% Idioma no reconocido

Nombre del documento: Tesis_Castillo_Chuquitarco-Antiplagio-Compilation_15-02-2023_.docx
ID del documento: 2a158a980ea13aca9544f3a67c96ea7364b00046
Tamaño del documento original: 3,52 Mo

Depositante: JOSÉ LUIS CARRILLO
Fecha de depósito: 15/2/2023
Tipo de carga: interface
fecha de fin de análisis: 15/2/2023

Número de palabras: 25.766
Número de caracteres: 169.915

Ubicación de las similitudes en el documento:

Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Tesina_Re-ID_Facial_Textura_Diego-Jose-08-02-2023.docx | Tesina_Re-ID_Fac... #b93932 El documento proviene de mi biblioteca de referencias 13 fuentes similares	2%		Palabras idénticas : 2% (479 palabras)
2	Tesina_Re-ID_Perez_Velastegui-08-02-2023.docx | Tesina_Re-ID_Perez_Velast... #ea5ca0 El documento proviene de mi biblioteca de referencias 10 fuentes similares	1%		Palabras idénticas : 1% (305 palabras)
3	Tesina_Re-ID_Lignia_Pichucho-08-02-2023.docx | Tesina_Re-ID_Lignia_Pichuc... #3fa6e0 El documento proviene de mi biblioteca de referencias 2 fuentes similares	1%		Palabras idénticas : 1% (282 palabras)
4	Tesina_Re-ID_Segovia-Echeverria-08-02-2023.docx | Tesina_Re-ID_Segovia-E... #ae664e El documento proviene de mi biblioteca de referencias 2 fuentes similares	< 1%		Palabras idénticas : < 1% (134 palabras)
5	Tesina_Re-ID_Vasquez_Vega-08-02-2023.docx | Tesina_Re-ID_Vasquez_Vega-... #d7e0e5 El documento proviene de mi biblioteca de referencias 1 fuente similar	< 1%		Palabras idénticas : < 1% (102 palabras)

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	doi.org | From Interface Mockups to Web Application Models | SpringerLink https://doi.org/10.1007/978-3-642-24434-6_20	< 1%		Palabras idénticas : < 1% (37 palabras)
2	Documento de otro usuario #45e065 El documento proviene de otro grupo	< 1%		Palabras idénticas : < 1% (20 palabras)
3	arxiv.org | Identifying Malicious Web Domains Using Machine Learning Techniques ... http://arxiv.org/abs/1902.08792	< 1%		Palabras idénticas : < 1% (17 palabras)
4	hdl.handle.net | Extracción de perfiles de alumnos de programación en estudios sup... http://hdl.handle.net/10045/128683	< 1%		Palabras idénticas : < 1% (17 palabras)
5	repositorio.upse.edu.ec | Una revisión del aprendizaje profundo aplicado a la cibers... https://repositorio.upse.edu.ec/bitstream/46000/8231/1/UPSE-RCT-2022-Vol.9-No.1-007.pdf	< 1%		Palabras idénticas : < 1% (13 palabras)

Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://www.youtube.com
2 https://www.google.com
3 https://amezoon.whymadeeasy.com/jp.php
4 https://doi.org/10.14569/IJACSA.2015.060927
5 https://doi.org/10.14445/22312803/IJCTT-V48P126

.....
Ing. Carrillo Medina, José Luis Ph.D
C.C.: 0501553788

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Certificación

Certifico que el trabajo de integración curricular: "**Sistema de detección de intrusos en sitios web, usando modelos y/o algoritmos de Machine Learning: caso práctico Phishing Google Chrome**" fue realizado por la señorita **Castillo Veloz, Mishell Estefanía** y el señor **Chuquitarco Velasco, Kevin Jair,** 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 prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 03 de febrero de 2023

Firma:

.....

Ing. Carrillo Medina, José Luis, Ph.D

C. C.: 0501553788

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Responsabilidad de Autoría

Nosotros, **Castillo Veloz, Mishell Estefanía** con cédula de ciudadanía N° 1728165406 y **Chuquitarco Velasco, Kevin Jair**, con cédula de ciudadanía N° 0550111595, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **"Sistema de detección de intrusos en sitios web, usando modelos y/o algoritmos de Machine Learning: caso práctico Phishing Google Chrome"** es de 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, 03 de febrero de 2023

Firmas:

.....

Castillo Veloz, Mishell Estefanía

C. C.: 1728165406

.....

Chuquitarco Velasco, Kevin Jair

C. C.: 0550111595

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Autorización de Publicación

Nosotros, **Castillo Veloz, Mishell Estefanía** con cédula de ciudadanía N° 1728165406 y **Chuquitarco Velasco, Kevin Jair**, con cédula de ciudadanía N° 0550111595, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Título: "Sistema de detección de intrusos en sitios web, usando modelos y/o algoritmos de Machine Learning: caso práctico Phishing Google Chrome"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 03 de febrero de 2023

Firmas:

.....	.....
Castillo Veloz, Mishell Estefanía	**Chuquitarco Velasco, Kevin Jair**
C. C.: 1728165406	C. C.: 0550111595

Dedicatoria

Quiero dedicar este proyecto a mis queridos padres, Rodrigo y Nuvia, por su amor incondicional y su apoyo constante a lo largo de todos estos años. Sin su confianza y dedicación a mi educación, no habría podido alcanzar este logro. De igual modo, a mi abuelita María, que a pesar de ya no encontrarse con nosotros siempre me brindo su compañía y valiosas enseñanzas. Este trabajo es una muestra de mi gratitud por todo lo que han hecho por mí.

Además, quiero dedicar a mis hermanos por ser una fuente constante de inspiración, motivación y apoyo, que me han ayudado a superar los desafíos y lograr mis metas.

También quiero dedicar este trabajo al ingeniero Diego Velasco, quien me motivó a continuar en esta carrera universitaria y me mostró las maravillosas oportunidades que esta profesión me ofrece. Su ejemplo y experiencia han sido una inspiración para mí, y me ha ayudado a mantenerme enfocado en mis metas.

Por último, quiero dedicar este proyecto al ingeniero José Carrillo, por su valiosa orientación y asesoramiento en el desarrollo de este trabajo. Agradezco sinceramente la oportunidad de haber trabajado bajo su dirección y estoy agradecida por la enseñanza y experiencia que me ha brindado.

Mishell Estefanía Castillo Veloz

Ecuador, enero de 2023

Dedicatoria

El presente proyecto de titulación realizado con mucho esfuerzo, se lo dedico con mucho amor a toda mi familia, en especial a mi papi Efraín y mi mami Paulina, que fueron las personas que desde pequeño me enseñaron buenos valores para ser una persona de bien, por todo el sacrificio realizado en estos años, por enseñarme a afrontar las dificultades de la vida y nunca rendirme, gracias a ustedes estoy cumpliendo mis sueños. También quiero dedicar este trabajo a mi hermana Gissela, por estar siempre motivándome y brindándome su ayuda sin pedir nada a cambio. Por último, a todas las personas que creyeron en mí, y confiaron en mi potencial.

Kevin Jair Chuquitarco Velasco

Ecuador, febrero 2023

Agradecimiento

Quiero expresar mi sincero agradecimiento a Dios por su constante guía y protección en todo momento, por darme la sabiduría y la paciencia para culminar con el presente trabajo.

También quiero agradecer a mis amados padres, Rodrigo y Nuvia, por su amor incondicional y su apoyo constante. Su confianza en mí y su dedicación a mi educación han sido fundamentales para llegar hasta aquí. No podría haberlo hecho sin ellos.

Así como también, agradezco a mis hermanos, por su apoyo y preocupación por mi bienestar emocional y físico durante el transcurso de mi vida académica, por su confianza en mis habilidades y por creer en mí siempre, incluso en los momentos difíciles.

Agradezco a mi amigo Kevin por su amistad a lo largo de mi vida universitaria. Su perseverancia y dedicación para terminar con el desarrollo del presente proyecto, además de su apoyo emocional en todo momento.

Finalmente, agradezco al ingeniero José Carrillo por su valiosa orientación y asesoramiento en el desarrollo de este proyecto. Su experiencia, conocimiento y paciencia fueron esenciales para la elaboración de este trabajo.

Mishell Estefanía Castillo Veloz

Ecuador, enero de 2023

Agradecimiento

Primeramente, agradezco a Dios por haberme permitido vivir esta oportunidad en mi vida, también por haberme guiado y darme la fortaleza para seguir adelante, a pesar de las dificultades presentadas en el camino. A mi papi Efraín, mi mami Paulina, les agradezco de todo corazón por siempre estar conmigo apoyándome, confiando en mí, por nunca negarme sus enseñanzas, por sus consejos y apoyarme con los recursos necesarios para estudiar. También agradezco a mi hermana Gissela, por los momentos divertidos que vivimos juntos, por las palabras de ánimo y nunca negarme su ayuda. A mis abuelitos, Gonzalo y Juana, mi tía Mayra por todos sus consejos y fuerzas que me daban para continuar. Agradezco a mi amiga y compañera del presente proyecto, por todo el esfuerzo que realizamos para cumplir este y muchos proyectos desarrollados en el transcurso de la carrera. A demás agradezco a todos mis amigos por los ánimos y momentos divertidos vividos. Finalmente agradezco a mi tutor de tesis Dr. José Luis Carrillo Medina, por las enseñanzas brindadas, por dirigirnos en el proceso de titulación y ser una persona de ejemplo en la que podemos confiar.

Kevin Jair Chuquitarco Velasco

Ecuador, febrero 2023

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Resumen

El internet a tomado fuerza en los últimos años, siendo una herramienta indispensable para realizar las tareas del día a día, lamentablemente estamos en una sociedad donde existen personas que buscan oportunidades de causar daño, este es el caso de los phisher, que realizan actividades maliciosas con el fin de engañar a sus víctimas para robar su información personal y así obtener acceso a una serie de plataformas, ya sean bancarias o personales. Los phishers engañan a sus víctimas con la creación de sitios web falsos que son similares a los sitios web verdaderos, usando un formulario, donde las víctimas ingresan sus datos y seguido de ello son robados. Los algoritmos de Machine Learning son usados en la ciberseguridad por los buenos resultados que muestran al detectar anomalías informáticas, por esta razón este proyecto se hace uso de modelos y/o algoritmos de Machine Learning para el desarrollo de un Sistema de Detección de Intrusos (IDS) para la detección de sitios web con phishing siguiendo la metodología Scrum. La aplicación fue probada y validada, tanto en un ambiente de entrenamiento como en un ambiente real/simulado, con ayuda del simulador de phishing Zphisher obteniendo resultados aceptables que están en el rango de la revisión de la literatura realizada al iniciar este proyecto.

Palabras clave: Machine Learning, sistema de detección de intrusos, phishing, sitios web.

Abstract

The Internet has gained strength in recent years, being an indispensable tool to perform daily tasks, unfortunately we are in a society where there are people looking for opportunities to cause harm, this is the case of phishers, who perform malicious activities in order to deceive their victims to steal their personal information and thus gain access to a number of platforms, whether banking or personal. Phishers deceive their victims by creating fake websites that are similar to real websites, using a form, where victims enter their data and then their data is stolen. Machine Learning algorithms are used in cybersecurity because of the good results they show when detecting computer anomalies, for this reason this project makes use of Machine Learning models and/or algorithms for the development of an Intrusion Detection System (IDS) for the detection of phishing websites following the Scrum methodology. The application was tested and validated, in a training environment and a real/simulated environment, the Zphisher phishing simulator was used, obtaining acceptable results that are in the range of the literature review made at the start of this project.

Key words: Machine Learning, intrusion detection system, phishing, websites.

Capítulo I

Introducción

Propósito y contextualización del tema

En la actualidad, un gran número de actividades cotidianas se han trasladado al Ciber-espacio, a través de servicios digitales, como el ir de compras y/o realizar movimientos bancarios, entre otras. Más aún, en esta época de pandemia Covid-19 ha aumentado la demanda de determinados productos lo que ha hecho que los canales digitales cobren hoy más importancia de la que ya se tenían en un mundo globalizado, esta ha provocado un rápido desarrollo del Internet y de las tecnologías de comunicación (Sahingoz et al., 2019) que permiten suplir con las necesidades de acceso de los usuarios a distintas plataformas web, que soportan este tipo de servicios. Como consecuencia, el Internet se ha convertido en un lugar de almacenamiento de grandes cantidades de información de todo tipo. Esto ha provocado que, este tipo de plataformas se conviertan en el principal objetivo de ciber-ataques (Sahingoz et al., 2019).

En general, el Internet es una infraestructura no controlada y con poca seguridad, que presenta un conjunto inmenso de vulnerabilidades a los usuarios, sistemas e infraestructura física, amenazas que pueden ser aprovechadas para generar daños financieros, robos de identidad, e incluso pérdida de usuarios como es el caso de comercio electrónico (Mohammad et al., 2014; Sahingoz et al., 2019), entre otros.

Algunas de estas amenazas están presentes en Internet, a las cuales se les conocen como amenazas cibernéticas y estas pueden ser: Phishing, Malware, Botnes, Ramsomware, Cryptojacking, Brute Force Attack. Una de las principales y más grandes amenazas, existentes al día de hoy, es el Phishing, que están presentes en sitios web (Anupam & Kar, 2021). Phishing es considerado como uno de los ciberataques más sencillo, pero uno de los más efectivos. Phishing se define como una técnica de Ingeniería Social que pretende hacerse

pasar por una entidad honesta, con el fin de obtener información privada de un usuario y posiblemente utilizarla con fines ilegales (Hr et al., 2020a; Mohammad et al., 2014; Sönmez et al., 2018).

Según el informe de CISCO sobre las tendencias de amenazas de seguridad, en el año 2021, indica que alrededor de un 90% de violaciones de datos han sido producidos debido a ataques de Phishing.(Apps, 2022). Por otro lado, el informe Cost of a Data Breach Report de IBM 2021 reveló que los ataques de Phishing son considerados como el segundo vector de ataque más costoso para las organizaciones. Para el año 2020, este tipo de ataques costo un valor aproximado de $3,86 millones de dólares y para el 2021 costo $4,65 millones de dólares (Apps, 2022). Esto refleja que el impacto al sufrir un ataque phishing, ya sea individual o dentro de una organización, puede llegar a causar una gran pérdida económica.

En este contexto, la seguridad informática es un factor clave para resguardar la información tanto privada como financiera de usuarios y de empresas (López Cruces, 2016). En la actualidad, se han realizado estudios e investigaciones sobre como 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, con el desarrollo de la tecnología los sitios web con Phishing se han ido perfeccionando con el tiempo, lo que demanda el desarrollo de sistemas de detección de Phishing que sean capaces de distinguir patrones (características) que permitan distinguir sitios web con Phishing de aquellos que son legítimos (Coronado Huamán et al., 2020). Este proceso debería ser realizado de forma constante el cual lleva mucho tiempo de forma manual, por lo que la mejor opción es optar por la implementación de sistemas de detección automáticos, los cuales utilizan técnicas de Inteligencia Artificial, específicamente Machine Learning.

El propósito del presente proyecto es desarrollar un Sistema de Detección de Intrusos (IDS) enfocado a la detección de sitios web con Phishing, como una extensión de Google

Chrome, para lo cual se pretende utilizar modelos y/o algoritmos de Machine Learning (ML), con la intención de automatizar el proceso de identificación de sitios web con Phishing, así como también brindar una precisión alta de detección. Finalmente, esta extensión podría ser una posible solución contra la problemática de robos de información que se producen actualmente mediante sitios web con Phishing.

Justificación

En los últimos años, el Internet ha tenido un crecimiento notorio debido a la transformación que ha tenido en el desarrollo de actividades referentes al trabajo, la educación y/u otras tareas comunes en la vida cotidiana. Actividades que en general realizan algún tipo de transacción, al enviar y/o recibir datos y/o información, debido al intercambio de información se pueden presentar vulnerabilidades y/o riesgos como la perdida de datos privados, fraude de datos, daños monetarios, robos de cuentas, e incluso, una vez obtenido nuestros datos privados, los piratas informáticos pueden ser capaces de suplantar nuestra identidad (Karabatak & Mustafa, 2018). Es decir, que el uso del Internet sin ningún tipo de seguridad tiene una gran cantidad de amenazas. Estas amenazas por lo general son aprovechadas por personas mal intencionadas (hackers) que además de generar perdidas 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 sencillos y efectivos para el robo de información (Aguilar, 2017)**.** Dado el éxito de este tipo de ataque, los phishers cada vez los incrementan, siendo uno de estos aquellos direccionados hacia los motores de búsqueda. Estos consisten en crear un sitio web con Phishing que se convierte en el principal resultado al realizar una búsqueda en un motor de búsqueda y cuando el usuario cae en esta trampa los hackers robarán sus datos y/o información cuando navega por la página web falsa (Ali & Malebary, 2020; Chiew et al., 2018).

Por ende, un Sistema de Detección de Intrusos (IDS) para la detección de Phishing en sitios web, vendría a ser de suma importancia al precautelar la seguridad de los datos de usuario mientras navega por internet. En este trabajo se plantea el desarrollo de este IDS el cual será implementado cómo una extensión para el navegador Google Chrome, debido a que estas son fáciles de instalar, que están al alcance de cualquier usuario y que están inmersos en el navegador.

Además, con ayuda de modelos y/o algoritmos de Machine Learning se pretende implementar un modelo que realice la predicción de los sitios web con Phishing de una manera eficiente, mediante la extracción de una serie de características a partir de una URL, las mismas que serán determinadas mediante un estudio y análisis de la literatura científica sobre estas dando énfasis a las más relevantes y que permitan mejorar las predicciones del IDS.

Objetivos

Objetivo general

Desarrollar un sistema de detección de intrusos en sitios web, usando modelos y/o algoritmos de Machine Learning: Caso Práctico Phishing Google Chrome

Objetivos específicos

- Conocer el estado del arte sobre métodos y técnicas para la detección de intrusos en sitios web, basado en phishing por motores de búsqueda - Google Chrome.
- Implementar un sistema de detección de intrusos en sitios web, a través del desarrollo de una extensión para Google Chrome, empleando técnicas de Machine Learning.
- Validar los resultados, analizar los errores y ajustar los modelos del sistema de detección de intrusos.

Metodología

El presente proyecto tiene como meta desarrollar un sistema de detección de intrusos capaz de detectar sitios web con Phishing y así, cumplir con los objetivos planteados. La metodología que seguirá este proyecto consta de 3 fases:

En la fase I se analiza la literatura científica relacionada con el objeto de estudio para formular el marco teórico de este trabajo, para lo cual se utilizarán varios métodos teóricos correspondientes a la investigación científica, como el método histórico-lógico y método análisis-síntesis. Se analiza las características de los sitios web, es decir, se determina los recursos de comprobación que permiten determinar si un sitio web tiene o no Phishing, a partir de una URL y además se estudia los modelos y/o algoritmos de Machine Learning (ML) más utilizados, que presentan una mayor precisión y los de mayor frecuencia para la detección de Phishing en sitios web. Esto se realizará utilizando la base de datos bibliográfica SCOPUS y la plataforma de Data Science KAGGLE.

La fase II fase consiste en el desarrollo del sistema de detección de intrusos y de una extensión de Google Chrome, para lo cual, se usarán estándares, técnicas y metodologías correspondientes al desarrollo de software. En primer lugar, se selecciona datasets que posean todas las características seleccionadas. Luego se crea varios escenarios, variando la cantidad de características, de sitios web legítimos y con phishing, así como combinación de escenarios. A continuación, se implementan los modelos y/o algoritmos de Machine Learning seleccionados en la fase anterior. Para luego, realizar las pruebas en cada escenario con cada modelo de Machine Learning implementado. Para luego, de acuerdo con los resultados realizar los ajustes respectivos. Además, se crea un dataset propio a partir de un dataset que contiene URL’s legítimas y URL’s con phishing (obtenido desde un repositorio de datos), utilizando el código de extracción de características desarrollado anteriormente, esto con el fin de entrenar al modelo de predicción con datos actuales y realizar pruebas al modelo de predicción mediante la utilización de un simulador de sitios web con phishing. Y al final se implementa y se despliega

una extensión de Google Chrome, que será capaz de realizar peticiones al modelo de predicción desarrollado y cargado a la nube.

En la última fase se pretende evaluar el IDS, para lo cual se emplearán los métodos experimental y empírico, con el fin de corroborar los resultados obtenidos una vez el sistema de detección de intrusos es implementado y desplegado hacia el público.

Capítulo II

Marco Teórico

En este capítulo se realiza un estudio teórico sobre los sistemas de detección de intrusos (IDS), los recursos de comprobación (características), modelos y/o algoritmos de Machine Learning, métricas de precisión, que son los elementos esenciales para el desarrollo de un sistema de detección de phishing para determinar si un sitio web contiene phishing o no.

Para el desarrollo de esta temática se realizó una pequeña revisión de la literatura, para lo cual se utilizó la base de datos bibliográfica de resúmenes y citas de artículos de revistas llamada SCOPUS. En primer lugar, se generó una cadena de búsqueda con los términos relacionados con el objeto de estudio propuesto. Luego se procesó la cadena de búsqueda en SCOPUS y se procedió a revisar y seleccionar artículos relevantes para la investigación basándonos principalmente en 3 criterios i) los artículos deben relacionarse directamente con el tema de investigación, ii) la cantidad de veces que el artículo ha sido citado sea mayor o igual a 7 (en promedio) y iii) que los artículos se encuentren comprendidos entre el período del 2017 al 2022. De acuerdo con estos criterios se encontraron 11 artículos, la información correspondiente se presenta en el Anexo 1: Artículos para la revisión de la literatura (SCOPUS). Con el fin de obtener una mejor comprensión sobre las características que son utilizadas con mayor frecuencia en la detección de phishing, en sitios web, se utilizó la plataforma de Data Science KAGGLE para buscar dataset relacionados con la temática de investigación. Y después de ingresar las siguientes palabras claves: i) features, ii) phishing y iii) detection, se obtuvieron un total de 11 dataset, tal como se puede apreciar en el Anexo 2: Datasets seleccionados en la revisión de la literatura implementada. Además, al final de este capítulo se enmarcan la información sobre extensiones Google Chrome, con el fin de abarcar totalmente la temática de este trabajo.

Sistemas de detección de intrusos (IDS)

Un Sistema de Detección de Intrusos (IDS) es un sistema de software o hardware que tiene como objetivo identificar actividades maliciosas en los sistemas informáticos y así mantener la seguridad de los usuarios (Khraisat & Alazab, 2021). Existen dos tipos de IDS: el primero es un sistema de detección de intrusos basados en firmas (SIDS) el cual tiene como objetivo encontrar un ataque conocido, mediante la comparación de patrones que previamente ya han sido encontrados y registrados en una base de datos (Khraisat et al., 2018); el segundo tipo de IDS es un Sistema de detección de intrusos basado en anomalías (AIDS), el cual se basa en encontrar comportamientos anormales mediante el análisis del flujo de la red, no contiene una base de datos como los SIDS y es muy útil para la identificación de ataques no conocidos (Alazab et al., 2012).

Un IDS requiere de una o varias entradas para poder detectar algún tipo de ataque, estas son conocidas como recursos de comprobación y específicamente para este proyecto son las características de los Sitios Web que se puedan extraer a partir de una URL y que se encuentran almacenados en un dataset. Actualmente, es uno de los enfoques más comunes que están siendo utilizados para enfrentar ataques a la seguridad contra las redes informáticas, de hoy día. Debido a la constante evolución del comportamiento de los ataques, se ha implementado las técnicas de aprendizaje automático (ML) a los IDS, con el fin de mejorar la precisión en la detección de ataques (Bui et al., 2021). Los tipos de ataques que detecta un IDS pueden ser de distinta naturaleza, entre los que se destaca está el malware, por ejemplo: Ransomware, Adware, Spyware, etc., también detecta ataques de ingeniería social por ejemplo: Phishing, Span, Scareware, entre otros (Dushimimana et al., 2020). En la Actualidad, el ataque más usado por los piratas informáticos es el phishing, debido a que es un ataque fácil de hacerlo y muy efectivo. Este tipo de ataque ha provocado pérdidas económicas y de información que ha afectado a las personas y empresas. Los piratas informáticos lo primero que realizan es engañar a la víctima redirigiendo a sitios web falsos, para luego hacer que la

persona ingrese sus datos, y así obtener información privada importante, con el fin de robar dinero y/o su identidad (Aguilar, 2017).

Características para detección de intrusos – Phishing

De acuerdo, con el estudio de la literatura realizado se recogieron un total de 62 características, registradas en el Anexo 3: Tabla de características para la detección de Phishing. Cabe mencionar que, se encontraron gran cantidad de características demasiado específicas, por lo que estas fueron globalizadas como se puede observar en el Anexo 2: Datasets seleccionados en la revisión de la literatura implementada. Al final, se seleccionó un total de 30 características, basados en la frecuencia con la que aparecen y esta debe ser mayor o igual al valor de la media de la frecuencia total. Las características seleccionadas se muestran ordenadas de forma descendente, con respecto a la frecuencia, como se muestra en la Tabla 1: Características de sitios web a partir de una URL, así como también, se presenta una breve descripción de cada recurso de comprobación (característica) y el número de artículo en el que se presenta la característica, teniendo como referencia el orden que se muestra en el Anexo 1.

Tabla 1

Características de sitios web a partir de una URL

Ord.	Recurso de comprobación	Descripción	Artículos
1	Presencia de dirección IP	La utilización de una dirección IP en lugar del nombre de dominio en una URL incrementa la posibilidad de que el sitio web sea víctima de phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018) .	1,3,4,5, 6,7,8,9, 10,11

Ord.	Recurso de comprobación	Descripción	Artículos
2	Longitud de la URL	Los sitios web de phishing pretenden ocultar el nombre del dominio, por lo que usan URL largas. La longitud de la URL de un sitio web legítimo es menor a los 54 caracteres, si se excede hasta los 75 caracteres tiene una alta probabilidad de considerarse phishing, y si es igual o supera los 75 caracteres es considerado phishing en su totalidad (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,8,9, 10,11
3	Presencia del símbolo @	El símbolo (@) en una URL, logra que el navegador web ignore todo aquello antes de dicho símbolo y esto aumenta la probabilidad de redirigir a sitios web con phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,8,9, 10,11
4	Estado SSL	Se enfoca en conocer si el sitio web posee un certificado SSL, debido a que los sitios web con phishing no encriptan los datos enviados, por lo que, no los tienen. Para identificar, una URL inicia con HTTPS y sus proveedores son confiables (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz	1,2,4,5, 6,7,8,10

Ord.	Recurso de comprobación	Descripción	Artículos
		et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	
5	Edad del dominio	Un sitio web puede ser valorado por la duración de su dominio. Si el tiempo desde su creación es menor a 6 meses tiene una alta probabilidad de ser un sitio web con phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,8,9, 10,11
6	Redirección de doble barra	Si la posición de (//) es mayor a 7, es considerado como un sitio web de phishing. Eso porque en los sitios legítimos, se utiliza una sola vez la redirección de doble barra (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,8,9, 10,11
7	URL de anclaje	Consiste en contar la cantidad de veces que las etiquetas <a> con enlaces del sitio web dirigen a un dominio diferente de este. Si la cantidad excede el valor de 31 % es considerada como sospechosa de phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,8,9, 10,11
8	Prefijo / Sufijo	El uso de prefijos y sufijos en URL de sitios web con phishing se consiguen utilizando el símbolo (-), por lo	1,3,4,5, 6,7,8,9,

Ord.	Recurso de comprobación	Descripción	Artículos
		que, los legítimos no lo utilizan (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	10,11
9	Enlaces en etiquetas	Se examinan todas las etiquetas existentes dentro del sitio web y a donde se redirigen. En caso de que, una de las etiquetas redirige a un sitio web de la lista negra se considera phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,5,6, 7,8,9,11
10	Deshabilitar clic derecho	En la actualidad, en los sitios web legítimos se encuentra desactivada la opción de clic derecho, para evitar que los usuarios puedan realizar cambios en el código fuente del sitio (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,10, 11
11	Uso de ventana emergente	Las ventanas emergentes aparecen con un menú en la pantalla, para después desaparecer con un clic. Una señal de un sitio web con phishing es cuando este solicita información al usuario a través de las ventanas emergentes (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,10, 11

Ord.	Recurso de comprobación	Descripción	Artículos
12	Favicon	Favicon es el ícono que se usa para identificar algún sitio web de manera fácil. Si el Favicon de un sitio web es diferente al dominio que se muestra en la URL, entonces tiene una alta probabilidad de presencia de phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,10, 11
13	URL anormal	Se revisa si la URL contiene hostname y que esta coincida con el dominio en la URL, en caso de no poseer estas 2 características, entonces es considerado como phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,6, 7,8,9,11
14	IFrame	Iframe son etiquetas que son utilizadas para redirigir a otro sitio web dentro de un mismo sitio web. Este tipo de etiqueta puede ser utilizada para engañar a los usuarios (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,11
15	Registro DNS	El registro DNS de un sitio web proporciona una gran cantidad de información importante, por lo que, los sitios web con phishing ocultan este registro (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018;	1,3,4,5, 6,7,9,10, 11

Ord.	Recurso de comprobación	Descripción	Artículos
		Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	
16	Índice de Google	Un sitio web con phishing no es indexado por Google debido a su corta vida útil (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,11
17	Puerto utilizado	Los puertos que son considerados como confiables son: el 8080 y el 443. Si el sitio web utiliza uno diferente, tiene alta probabilidad de ser phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,5,6, 7,8,9,11
18	Request URL	Examina si los objetos externos contenidos en una página web se cargan desde otro dominio. Si la dirección de la URL se encuentra fuera del dominio, entonces es considerado como phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,10, 11
19	SFH (Controlador de formulario de servidor)	Se enfoca en la gestión de formularios y revisa si el botón submit retorna un mensaje vacío una vez completado, o también no retorna ninguna respuesta para etiquetas a un sitio web como con phishing (Alam	1,3,4,5, 6,7,9,11

Ord.	Recurso de comprobación	Descripción	Artículos
		et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	
20	Recuento de redirección del sitio web	Consiste en las veces que las fuentes redirigen a una sola dirección web o también, a un sitio web con un dominio diferente al que se muestra en la barra de búsqueda (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,11
21	MouseOver	Esta función permitía mostrar información sobre el sistema en la parte inferior de la pantalla. Sin embargo, actualmente, ya no se utiliza en la mayor cantidad de sitios web legítimos, por lo que, si un sitio web lo utiliza es considerado como sospechoso (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,4,6,7, 9,10,11
22	Tráfico web	Un sitio web puede ser evaluado de acuerdo con las visitas que recibe diario, semanal y/o mensual. Generalmente, mientras más alto sea este valor, el sitio se considera más fiable (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,11

Ord.	Recurso de comprobación	Descripción	Artículos
23	Servicio de Acortamiento	Un servicio de acortamiento de URL es una técnica que es implementada en una URL para abreviar y que esta sea capaz de dirigir a la misma página que la dirección original. La mayoría de los sitios web con phishing utilizan este tipo de servicios (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,5,6, 7,9,11
24	Duración del registro del dominio	Esta información se obtiene con el registro whois; si el número de años que ha sido registrado el dominio de un sitio web es menor o igual a 1 año es considerado como phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,5,6, 7,8,9,11
25	Token HTTPS	Se trata de la utilización del protocolo TLS/SSL juntamente con HTTP seguro (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,5,6,7, 9,11
26	Envío de información al correo electrónico	Se revisa si el sitio web utiliza algún tipo de servicio ‘mail() to’ dentro del mismo para ser considerado como phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,4,5,6, 7,9,11

Ord.	Recurso de comprobación	Descripción	Artículos
27	Rango de página	El rango de un sitio web se calcula contando los enlaces salientes y entrantes existentes en este, y este valor representa la importancia del mismo. Si este valor es menor que 0.2, entonces tiene altas sospechas de phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,11
28	Informe estadístico	Los informes estadísticos brindan información acerca de los sitios web legítimos y con phishing, además de otros datos estadísticos. (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,3,4,5, 6,7,9,11
29	Presencia de subdominio	Un sitio web de phishing contiene más de 2 subdominios en su URL. Para identificarlo, se debe observar la cantidad de puntos existentes en el dominio, ya que si es mayor a 2 se considera phishing (Alam et al., 2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018).	1,4,5,6, 7,9,11
30	Enlaces que apuntan a la página	La validez de un sitio web se puede calcular basándose en la cantidad de enlaces que apuntan a dicho sitio. Al menos deben existir 2 de estos enlaces para ser considerado como legítimo (Alam et al.,	1,5,6,7, 8,9,11

Ord.	Recurso de comprobación	Descripción	Artículos
		2020; Hr et al., 2020a; Jain & Gupta, 2018; Korkmaz et al., 2020; Lakshmi et al., 2021; Sönmez et al., 2018)	

Modelos y/o algoritmos de Machine Learning

Machine Learning (ML) o Aprendizaje Automático, se define como un campo de estudio que tiene como objetivo hacer que las computadoras tengan la capacidad de aprender, basándose en un conjunto de datos, para después poder tomar decisiones (predecir) por si sola sin la necesidad de estar programándolas (Mahesh, 2019; Ray, 2019; Song et al., 2017).

Machine Learning puede ser aplicado en varios campos de la ciencia cómo: la robótica, videojuegos, reconocimiento de patrones, minería de datos, procesamiento de lenguaje natural, medicina, seguridad informática, entre otros (Akinsola, 2017). Los algoritmos de Machine Learning hacen referencia a un conjunto de líneas de código, las cuales ayudan al análisis de un conjunto de datos, con el fin de crear un modelo que tenga la capacidad de predecir o clasificar información (Hao & Ho, 2019).

Los algoritmos y/o modelos de Machine Learning tiene diferentes formas de aprendizaje como son: el supervisado y el no supervisado. El aprendizaje supervisado se basa en el uso de datos etiquetados, con la finalidad que el modelo aprenda de los datos tras una serie de iteraciones. Los algoritmos habituales en el aprendizaje supervisado son: Árboles de decisión, Support Vector Machines (SVM), Naive Bayes, etc. Por otro lado, el aprendizaje no supervisado descubre patrones a partir de un conjunto de datos que no se encuentran etiquetados y para después clasificarlos. Algunos de los algoritmos más populares en el aprendizaje no supervisado son: KMeans, Hierarchical Clustering, DBSCAN. (Rajoub, 2020; Ray, 2019).

Los modelos y/o algoritmos de Machine Learning pueden ser de dos tipos: Regresión/Clasificación y Agrupación/Reducción. Los primeros se basan en aprendizaje supervisado, que trabajan directamente con los datos originales de entrada y salida, dando como resultado una mayor eficiencia en los procedimientos empleados en el entrenamiento y las pruebas (Yuan et al., 2012). Los segundos pertenecen a la categoría de agrupación/reducción los cuales se basa en aprendizaje no supervisado, y son considerados como una de las formas más comunes para realizar el análisis de datos (Gates & Ahn, 2017). Los datos usados en este tipo de modelos no tienen una salida definida (categoría) para indicarle a que grupo pertenece los datos de entrada, por esta razón, el modelo y/o algoritmo es quién busca patrones para agruparse en diferentes categorías.

En base a la revisión de literatura (ver Anexo 1: Artículos seleccionados en la revisión de la literatura) se identificó los algoritmos y/o modelos más utilizados para detectar Phishing. Además, se analizó la precisión máxima y la incidencia que tienen dichos modelos en la detección.

A continuación, en la Tabla 2: Modelos y/o algoritmos de Machine Learning, se describen los algoritmos y/o modelos encontrados seleccionados:

Tabla 2

Modelos y/o algoritmos de Machine Learning

Ord.	Modelo y/o Algoritmo	Descripción
1	Decision Tree	Decision Tree (Árbol de Decisión) es un algoritmo de aprendizaje supervisado, el cuál es usado para tareas de clasificación y regresión. Un árbol de decisión tiene una estructura jerárquica que clasifica los datos desde la raíz (nodo raíz) hasta los nodos hoja. Cada nodo del árbol de decisión representa una condición, mientras

Ord.	Modelo y/o Algoritmo	Descripción
		que cada nodo hoja representa una respuesta para la condición. La clasificación/regresión comienza en el nodo raíz y se va esparciendo hasta tener un árbol completo y a partir de ahí realiza la clasificación o regresión (Alzubi et al., 2018; Ray, 2019). Según la revisión de la literatura realizada Decision Tree alcanza una precisión máxima de 96,60% en la detección de phishing.
2	Random Forest	Random Forest (Bosque aleatorio) es uno de los algoritmos de aprendizaje supervisado más utilizados debido a su simplicidad y precisión. Es usado para la clasificación y regresión de información, además se basa en crear un sin número de árboles de decisión para obtener una salida que combina los árboles de decisión y obtener uno final. Random Forest consta de dos etapas, la primera es crear los bosques aleatorios, y segunda es realizar la predicción a partir del clasificador de bosque aleatorio (Alzubi et al., 2018). Random Forest alcanza una precisión máxima de 99,33% en la detección de phishing según la revisión de la literatura realizada.
3	Ada Boost	El algoritmo de Ada Bosost es un clasificador meta-estimador, es decir usa métodos para reducir la varianza de un estimador y mejorar su precisión. Lo primero que realiza este algoritmo es ajustar un clasificador con un conjunto de datos, luego realiza copias del clasificador y las ajusta con el mismo conjunto de datos. Si los pesos de los datos clasificados son incorrectos se ajustan para que los clasificadores puedan resolver en un futuro problemas más

Ord.	Modelo y/o Algoritmo	Descripción
		complejos (Freund & Schapire, 1997). En la detección de phishing Ada Boost alcanza una precisión máxima del 99,1% según la revisión de la literatura realizada.
4	Neural Networks	Neural Networks (NN) o Redes Neuronales en los últimos años, han recibido una gran acogida por parte de los investigadores, debido a que se los cataloga cómo el aprendizaje moderno ya que su desarrollo fue motivado por la funcionalidad del cerebro humano (Petersen, 2022). Una red neuronal tiene como objetivo enseñar a las computadoras a procesar los datos de forma cómo lo hace el cerebro humano, donde se usan nodos (neuronas) que están interconectadas, similar a como lo es el cerebro humano. Las Redes Neuronales aprenden de sus errores y de esos errores mejora continuamente, por lo que las redes neuronales son usadas para resolver problemas complejos para obtener una mayor precisión (Lauzon, 2012). Las Redes Neuronales alcanzan una precisión máxima del 97% en la detección de phishing según la revisión de la literatura realizada.
5	Support Vector Machines	El algoritmo Support Vector Machines (SVM) es un algoritmo de machine Learning de aprendizaje supervisado, que puede manejar casos tanto como de clasificación y de regresión. Su funcionamiento se basa en un plano en el espacio, el cual se va formando según las características de los datos de entrenamiento, seguido de ello los datos son categorizadas mediante un separador, para que al final el

Ord.	Modelo y/o Algoritmo	Descripción
		algoritmo pueda recibir nuevas características y predecir en la categoría a la que pertenecen (Ray, 2019). En la detección de phishing SVM alcanza una precisión máxima del 96,5% según la revisión de la literatura realizada.
6	Bagging	Bagging es un algoritmo de machine Learning que busca mejorar la precisión de algoritmos de clasificación débiles que son entrenados individualmente. Este algoritmo se basa en ajustar clasificadores en base a los datos originales, los cuales toma subconjuntos de datos y luego realiza predicciones en base a un promedio o votación. Bagging, es un meta estimador, que es usado para reducir la varianza de un clasificador, por ejemplo, puede ser aplicado a Random Forest, Decision Tree o cualquier otro clasificador (Khan et al., 2020). Los algoritmos combinados con bagging alcanzan una precisión máxima del 98,3% según la revisión de la literatura realizada.

Extensiones Google Chrome

Las extensiones de Google Chrome son aplicaciones que se ejecutan dentro del entorno de un Sitio Web, tienen como finalidad proporcionar nuevas funcionalidades, para mejorar la experiencia del usuario, debido a que la combinación de características del navegador es posible que el usuario realice muchas cosas al mismo tiempo (Mehta, 2016). El navegador Google Chrome implementó este tipo de funcionalidades desde el año 2010, es decir, a partir de la cuarta versión se pudo crear extensiones. Se han creado extensiones para diferentes navegadores cómo Opera, Brave, Mozilla Firefox y Microsoft Edge, pero no siempre

las extensiones de Google Chrome son compatibles con otros navegadores, debido a que Google se basa en Chromium, que es un proyecto de código abierto, mantenido por diversas compañías.

Algunas extensiones probadas y que dan buenos resultados en la detección de Phishing son: PIXM Phishing Protection, My Wot, Retruster Phishing Proction que pueden ser instalados en navegadores basados en Chromium y PhishWall instalado en Firefox, entre otros. Para más información acerca de las extensiones para detectar phishing se puede revisar el Anexo 4: Extensiones para la detección de phishing.

En este trabajo de investigación se propone desarrollar una extensión para Google Chrome la cual permitirá la detección de phishing en sitios Web, para lo cual se necesita tener conocimientos en tecnologías como HTML, CSS, JavaScript y JSON (Mehta, 2016). Las extensiones se pueden desarrollar desde cualquier sistema operativo y a comparación de las extensiones de otros navegadores las extensiones de Google Chrome tienen más acogida y uso, hasta el mes de septiembre del año 2022 Google Chrome es el navegador más popular con un 65,68% de uso (*Statcounter Global Stats - Browser, OS, Search Engine Including Mobile Usage Share*, s. f.).

Capítulo III

Implementación Del Sistema

En este capítulo se especifican todos los pasos que se realizaron para desarrollar el sistema propuesto, un sistema de detección de ataques tipos Phishing mediante el uso de modelos y/o algoritmos de Machine Learning, mismo que será instalado en Google Chrome a través de una extensión, la cual tendrá la funcionalidad de informar a un usuario si está en un sitio web legítimo o con phishing. En este capítulo, además, se especifica los pasos desarrollados para la creación de la API Rest.

Con el fin de tener una visión sobre cómo funcionará el sistema de detección propuesto, se explicará brevemente su proceso: primero se envía la URL del sitio web a la API, la API se encargará de extraer las características de acuerdo a la Tabla 1 (características seleccionadas), luego se realiza la predicción del sitio, respondiendo si el sitio web contiene phishing o no, mismo que será mostrado al usuario a través de la extensión Google Chrome desarrollada.

Para el desarrollo de la aplicación se utilizó la metodología de desarrollo de software ágil, debido a que este tipo de metodologías se caracterizan por tener interacciones continuas, las pruebas de software se aplican en todo el proceso de desarrollo de la aplicación (Srivastava et al., 2017). La metodología utilizada para el desarrollo del sistema es Scrum, porque es un framework que permite la gestión y el control del proceso de desarrollo de software (Srivastava et al., 2017). Se basa principalmente en la realización de varios Sprints durante el desarrollo. Cada Sprint es un periodo de tiempo donde se debe completar una funcionalidad del sistema, que dura entre 15 a 30 días máximo (Adi, 2015). Al finalizar un Sprint este debe proporcionar un entregable funcional, que será evaluado por el usuario antes de continuar con un siguiente Sprint.

Es necesario mencionar que Scrum cuenta con varios eventos que tiene como fin conservar un flujo regular durante el desarrollo del proyecto y prevenir posibles inconvenientes

y/o situaciones que puedan retardar la ejecución de los Sprints. Los eventos mencionados en (Ken & Sutherland, 2020), fueron aplicados y se explican a continuación:

1. **Sprint:** Es un periodo de tiempo, que varía entre 15 a 30 días máximo, y se desarrolla una funcionalidad y/o incremento del sistema. Un sprint debe entregar un producto funcional que aporte un valor tangible al sistema.
2. **Reunión de planificación del sprint (Sprint planning):** Se realiza en una reunión formal donde participan todos los miembros del equipo de desarrollo y tiene como objetivo organizar las historias de usuario que conformarán el sprint que va a ser desarrollado. Esta reunión no debe durar más de dos horas antes de cada ejecución de un sprint.
3. **Reuniones diarias (Daily Scrum):** Es una reunión que se realiza diariamente, con una duración máxima de 15 minutos y participan todos los miembros del equipo de desarrollo. Tiene como objetivo realizar un seguimiento del avance del sprint.
4. **Revisión del sprint (Sprint review):** Es una reunión que se realiza al finalizar el sprint, para revisar todo el trabajo realizado por el equipo de desarrollo y poder identificar oportunidades de mejora para el equipo y el desarrollo del proyecto.
5. **Retrospectiva del Sprint (Sprint Retrospective):** Planifica formas de aumentar la eficacia del siguiente Sprint, basados en la experiencia del Sprint anterior. El equipo Scrum discute los problemas, que se ha conseguido y que se espera conseguir, en base a estos se aplican los cambios para mejorar la efectividad del siguiente Sprint.

El siguiente punto a tener en cuenta se relaciona con las métricas de evaluación aplicadas a las diferentes modelos de Machine Learning que fueron probados para el desarrollo del presente proyecto, de acuerdo con (Xin et al., 2018) se definieron las siguientes métricas de evaluación:

Accuracy: Hace referencia al porcentaje de elementos correctamente clasificados entre positivos y negativos.

Precision: 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.

En la Tabla 3: Fórmulas de métricas de evaluación, se muestra las fórmulas aplicadas en la evaluación del modelo, mismas que están acordes con (Xin et al., 2018).

Tabla 3

Fórmulas de métricas de evaluación

MÉTRICA DE EVALUACIÓN	FÓRMULA
Accuracy	$accuracy = \frac{VP + VN}{VP + VN + FP + FN}$
Precision	$precision = \frac{VP}{VP + FP}$
Recall	$recall = \frac{VP}{VP + FN}$

En 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, para realizar la evaluación del modelo es necesario aplicar la matriz de confusión, que básicamente es una tabla en la cual se recogen todos los datos obtenidos de la

clasificación y permite distinguir entre clases (Goetz et al., 2015). En la Tabla 4: Matriz de confusión para Hunter Phisher, se muestra tal como se utilizará al momento de su aplicación.

Tabla 4

Matriz de confusión para Hunter Phisher

	POSITIVOS	**NEGATIVOS**
POSITIVOS	Phishing clasificados correctamente (VP)	Legítimos mal clasificados (FP)
NEGATIVOS	Phishing mal clasificados (FN)	Legítimos clasificados correctamente (VN)

Análisis y diseño del sistema

Análisis del sistema

De acuerdo a los lineamientos de la metodología Scrum, para la especificación de los requerimientos del sistema se utilizan Historias de Usuario (***HU***) y a partir de esta sección se involucran los miembros del sistema, como son: el propietario del proyecto o usuario del sistema (product owner), el desarrollador encargado de gestionar y controlar el equipo de desarrollo (scrum master) y el equipo de desarrollo (development team) (Kurnia et al., 2018). La designación de cada uno de estos roles se organizó como se indica en la Tabla 5: Rol de Scrum designados, en donde se encuentra el nombre del rol y del integrante del proyecto que tiene dicho cargo respectivamente.

Tabla 5

Rol de Scrum designados

N°.	**Rol**	**Integrante**	**Descripción**
01	Scrum Master	Kevin Jair Chuquitarco Velasco	Líder del equipo de Scrum

N°.	Rol	Integrante	Descripción
02	Product Owner	Dr. José Luis Carrillo Medina	Representa a las partes interesadas
03	Team Development	Mishell Estefanía Castillo Veloz Kevin Jair Chuquitarco Velasco	Desarrollo y diseño de la aplicación

De acuerdo, a la asignación de roles, indicadas en con la Tabla 5, es importante mencionar, que, dado que el proyecto se conforma únicamente de 2 integrantes, uno de ellos será el Scrum Master y también participará en las actividades correspondientes al team development. Con esta información se obtienen los requerimientos y la redacción de las historias de usuario. El Scrum Master del proyecto realiza la reunión inicial, en donde participan el usuario del sistema y el equipo de desarrollo. En la Tabla 6: Historias de usuario, se presentan las ***HU*** redactadas, en donde se especifica el usuario final del sistema como rol, las características y/o funcionalidades que se requieren respectivamente y la razón por la que debe ser implementando.

Tabla 6

Historias de usuario

ID	Nombre	Rol	Característica / Funcionalidad	Razón / Resultado
1	H.U. 01	Como usuario	Quiero que la extensión utilice el mejor algoritmo y/o modelo de Machine Learning para la detección de phishing en sitios web.	Para que la extensión realice predicciones con una buena precisión.
2	H.U. 02	Como usuario	Quiero un dataset que contenga características que permitan	Para entrenar el modelo de Machine Learning

ID	Nombre	Rol	Característica / Funcionalidad	Razón / Resultado
			identificar sitios web con phishing de los legítimos.	
3	H.U. 03	Como usuario	Quiero que el modelo de Machine Learning se encuentre almacenado en un servidor y pueda realizar predicciones a través de un servicio.	Para tener un servicio que pueda ser utilizado en otras aplicaciones.
4	H.U. 04	Como usuario	Quiero una extensión para el navegador Google Chrome que me informe si un sitio web contiene phishing.	Para determinar si estoy en un sitio web seguro mientras estoy navegando en la red con Google Chrome.

Luego de especificar las ***HU*** se realiza el Product Backlog del proyecto (lista ordenada de las historias de usuario redactadas inicialmente) (Kurnia et al., 2018), de acuerdo a la prioridad que se presentan en su implementación, las cuales constan en la planificación del proyecto.

En la Tabla 7: Product Backlog del proyecto, se propone y muestra las historias de usuario que se desarrollarán en el transcurso del proyecto con su respectiva estimación de tiempo en días, la fecha de inicio, la fecha de fin y el N° de Sprint al que corresponde cada historia de usuario especificada.

Tabla 7

Product Backlog del proyecto

Historia de usuario	Nombre	Estimación (días)	Fecha de inicio	Fecha de fin	N° de Sprint
1	H.U. 01	13	24/10/2022	09/11/2022	01

Historia de usuario	Nombre	Estimación (días)	Fecha de inicio	Fecha de fin	N° de Sprint
2	H.U. 02	20	10/11/2022	07/12/2022	02
3	H.U. 03	20	05/12/2022	30/12/2022	03
4	H.U. 04	8	02/01/2023	11/01/2023	04

Diseño del sistema

Esta sección abarca, en su totalidad, el diseño del sistema que se utiliza para cumplir con el objetivo principal del presente proyecto. Además, se presenta el diseño de la interfaz para crear la extensión Google Chrome propuesta, mediante la realización de mockups, o maquetas en español, que son bocetos que pretenden representar la Interfaz de Usuario (UI) probable para la aplicación a desarrollar, mostrando la apariencia general de la UI (Rivero et al., 2010). El sistema de detección de phishing implementado se denominó “Hunter Phisher”, su nombre hace referencia a un cazador (Hunter) de una persona con intenciones de robar información a través de una estafa (Phisher).

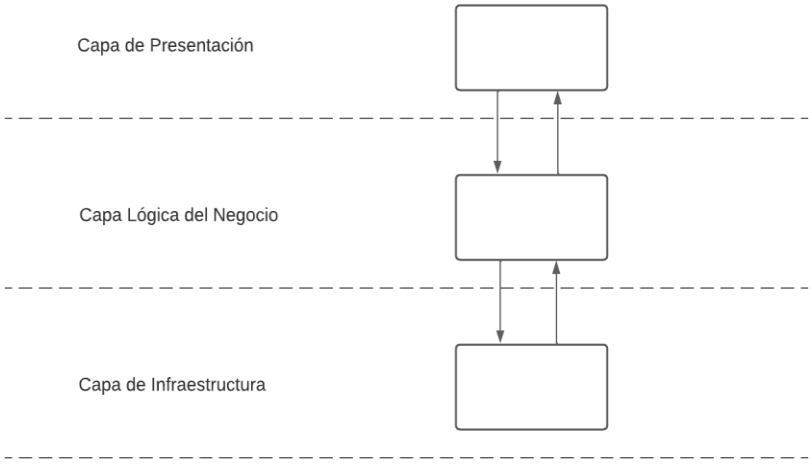
Diseño de arquitectura. De acuerdo con (Zahid et al., 2017) en su estudio Evolution in Software Architecture Recovery Techniques – A Survey señala que “La arquitectura de software describe los componentes del sistema de software y las conexiones entre ellos que dan lugar a un comportamiento especificado mediante la colaboración de los elementos de estos componentes”. Además, menciona que el diseño de la arquitectura es una parte fundamental para el desarrollo de un sistema.

Esta sección tiene como objetivo definir y diseñar la arquitectura de software que se implementa, así como también, las tecnologías que se utilizan para desarrollar el sistema propuesto.

Diagrama de la arquitectura lógica. Es importante la gestión de aplicaciones en capas, ya que ayuda a la separación de los archivos de la aplicación, dando como resultado un código más fácil para mantener y reutilizar. La extensión Google Chrome será desarrollada con el modelo de 3 capas, su diagrama se especifica en la Figura 1: Diagrama de la arquitectura lógica del sistema, donde se puede observar la capa de presentación, la capa lógica del negocio y la capa de infraestructura, las cuales interactúan entre sí con la finalidad de dar funcionamiento a la aplicación.

Figura 1

Diagrama de la arquitectura lógica del sistema



Definición de las tecnologías a usar

Una tecnología de desarrollo de software es un programa de software que es usado para construir diferentes tipos de aplicaciones. Existe un sinnúmero de tecnologías para el desarrollo de software, pero deben ser elegidas de acuerdo con el tipo de proyecto de desarrollo que se presente.

En la Figura 2: Diagrama de la arquitectura lógica del sistema con las tecnologías a usar, se presentan las tecnologías que se van a utilizar en cada capa perteneciente al modelo especificado en la Figura 1, para una mejor comprensión sobre el propósito que tiene cada

capa especificada anteriormente. A continuación, se procede a dar una breve explicación de cada una de estas tecnologías:

Capa de presentación:

La capa de presentación es la responsable de la interacción del usuario y la aplicación, se encarga de recolectar datos que son obtenidos a través de la interacción que surge entre la aplicación y el usuario, los datos se envían a las demás capas con el fin de procesar los datos y mostrar el resultado. A continuación, se especifican las tecnologías a usar en esta capa:

- Html (Lenguaje de Marcas de Hipertexto): Código que contiene una serie de etiquetas que encierran una serie de elementos, las cuales son usadas para estructurar y desplegar una página web (Gauchat, 2012).
- Css (Hojas de estilo en cascada): Es un lenguaje basado en reglas, las cuales permiten agregar estilos a un sitio web, con la finalidad de hacerlos más atractivos a los ojos del usuario final (Gómez, 2021).
- Js (JavaScript): Es un lenguaje de programación el cual permite implementar una serie de funcionalidades a un sitio web (Robbins, 2012).

Capa lógica del negocio:

La capa lógica del negocio contiene los programas encargados de recibir las peticiones de la capa de presentación, para luego enviarlas a la capa de infraestructura solicitando realizar la predicción correspondiente, y una vez retorne los resultados, esta capa se encargará de enviar la respuesta a la capa inicial o de presentación, para que la presente al usuario. La tecnología que se utilizará para esta capa es:

- Js (JavaScript): Es un lenguaje de programación de alto nivel, el cuál puede ser utilizado en diferentes sistemas operativos y sirve para el desarrollo de un sinnúmero de aplicaciones.

Capa de infraestructura:

Esta capa contiene los mecanismos necesarios para interactuar con la API propuesta en este proyecto y el modelo de Machine Learning entrenado. Esta capa está compuesta principalmente por: el modelo que estará entrenado y guardado como un SAV File; y un servicio que permitirá realizar la predicción de un sitio web, enviado a través de la extensión de Google Chrome. A continuación, se especifica las tecnologías a usar para desarrollar el servicio:

- Python: Es un lenguaje de programación de alto nivel, el cuál puede ser utilizado en diferentes sistemas operativos y sirve para el desarrollo de un sinnúmero de aplicaciones (FERNANDEZ, 2013).
- Flask: Es un “micro-framework” desarrollado en Python, se utiliza para el desarrollo de aplicaciones web de manera ágil y rápida (Grinberg, 2018).

Figura 2

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

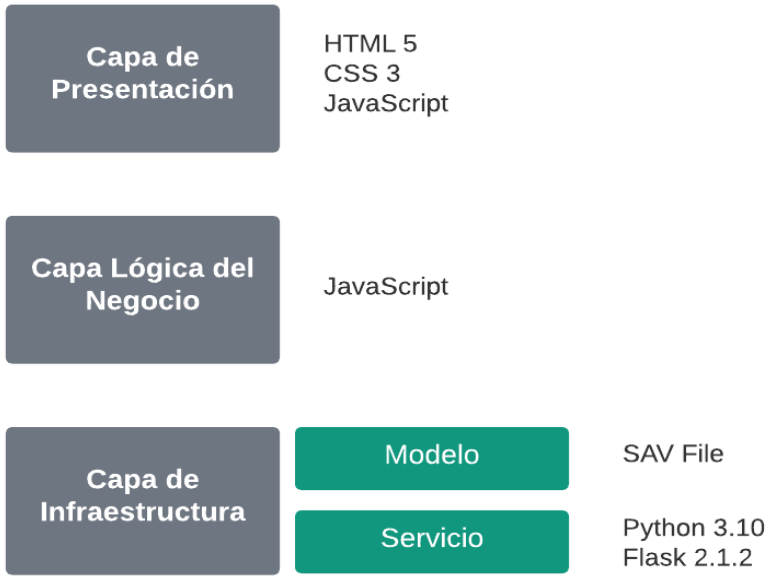
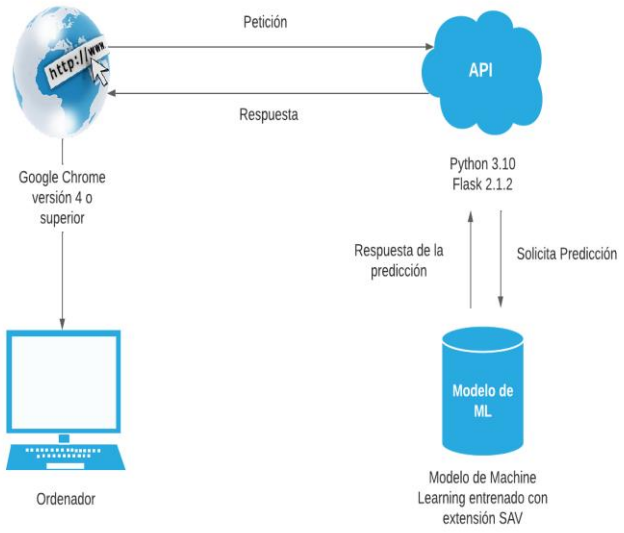


Diagrama de la arquitectura física. En la Figura 3: Diagrama de la arquitectura física del sistema, se presenta la arquitectura que se utilizará para el desarrollo de la extensión de Google Chrome (Hunter Phisher) a nivel físico, donde se especifica que se podrá acceder al sistema a través de cualquier ordenador que tenga instalado el navegador Google Chrome con una versión igual o superior a la 4.

Cuando el usuario se encuentra navegando por internet y de click en la extensión, se enviará la URL del sitio web actual, la misma llegará a la API, la cual extraerá las características de la URL y el sitio web, mismas que son la entrada al modelo de Machine Learning entrenado, que retornará si el sitio web es legítimo o tiene phishing y así la API pueda retornar una respuesta a la extensión.

Figura 3

Diagrama de la arquitectura física del sistema

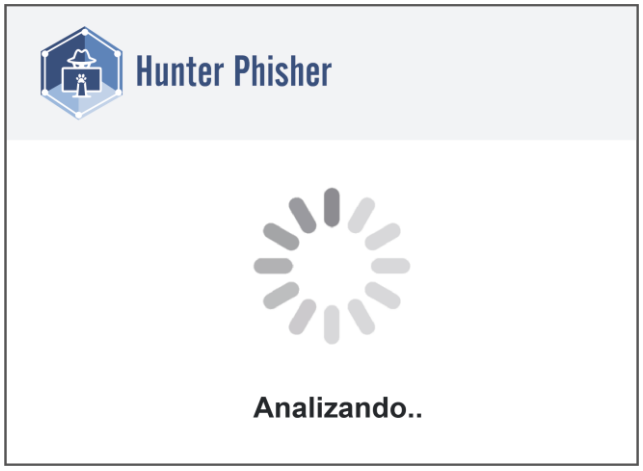


Mockups. Los Mockups son modelos ágiles usados para representar requisitos que son entendidos tanto como para los clientes y desarrolladores (Rivero et al., 2011). A continuación, se presentan los mockups para la aplicación Hunter Phisher.

- Mockup 1: La Figura 4: Mockup analizando sitio web, se presentará cuando se dé clic en la extensión e inicie a realizar la petición del servicio web.

Figura 4

Mockup analizando sitio web



- Mockup 2: La Figura 5: Mockup sitio web legítimo, se presentará cuando el sitio web sea legítimo.

Figura 5

Mockup sitio web legítimo

- Mockup 3: La Figura 6: Mockup sitio web phishing, se presentará cuando el sitio web contenga phishing.

Figura 6

Mockup sitio web phishing

Implementación de algoritmos y modelos de Machine Learning para sitios web Phishing.

La metodología Scrum indica que una vez realizado el Product Backlog del proyecto, en donde se especifican las historias de usuario y el número de Sprint al que corresponde, se realiza la planificación para cada Sprint, que se encuentran ordenados en base a la prioridad de desarrollo, es decir, se debe realizar un Sprint Backlog como corresponda, para después ejecutarlos como se planifico (Ken & Sutherland, 2020). Es importante mencionar que para llevar a cabo con las reuniones que señala Scrum se utilizó la herramienta Discord para realizar reuniones virtuales, y también se realizaron reuniones presenciales cuando se consideraron necesarias.

El hardware utilizado en este proyecto para la ejecución del código desarrollado durante todos los sprint fue un procesador Intel Core i5-1135G7 con 8 GB de memoria RAM y el sistema operativo Windows 11 Home.

Sprint 01: Selección del mejor modelo de Machine Learning

Para el desarrollo del primer Sprint, se tomó como base la Historia de Usuario H.U. 01 especificada en la Tabla 4, el cual indica que se debe seleccionar el mejor algoritmo y/o modelo de Machine Learning para la detección de sitios web con phishing.

Historias de usuario detalladas. La Tabla 8: Historia de usuario para la selección del modelo y/o algoritmo de Machine Learning, presenta la Historia de Usuario H.U. 01 del sistema de detección de phishing (Hunter Phisher) de forma detallada, donde se especifica los responsables del desarrollo, y los criterios de aceptación para la selección del mejor modelo y/o algoritmo para detectar phishing en sitios web.

Tabla 8

Historia de usuario para la selección del modelo y/o algoritmo de Machine Learning

Historias de Usuario	
Número: H.U. 01	**Usuario:** Usuario de internet
Nombre historia: Definición y selección del modelo para detección de sitios web con phishing**.**	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 13	**Interacción asignada:** 1
Programadores responsables: Mishell Castillo, Kevin Chuquitarco	
Descripción: ● Como usuario quiero que la extensión utilice el mejor algoritmo y/o modelo de Machine Learning para la detección de phishing en sitios web.	
Validación (Criterios de aceptación): ● Se implementarán los modelos y/o algoritmos de Machine Learning resultantes de una revisión de literatura que presenten valores de precisión (accuracy) superiores a la media calculada entre todos los resultados obtenidos. ● Se realizarán pruebas de cada modelo seleccionado y se registrarán sus resultados respectivamente. ● Se seleccionará el modelo y/o algoritmo de Machine Learning que tenga el mayor valor de accuracy.	

Sprint backlog. En la Tabla 9: Sprint backlog 01, se especifica las tareas que se realizaron para llevar a cabo el desarrollo del sprint, los responsables de realizar cada uno de estas, las respectivas fechas en las que se planificó ejecutar el sprint, el tiempo que se estimó en horas, el esfuerzo en horas que se trabajó realmente cada día y el estado en el que actualmente se encuentra cada tarea. Cabe mencionar que se presenta el sprint backlog ya finalizado.

Tabla 9

Sprint Backlog 01

	Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X
	1	24/10/2022	8 horas	09/11/2022	24/10/2022	25/10/2022	26/10/2022	27/10/2022	28/10/2022	31/10/2022	01/11/2022	02/11/2022	03/11/2022	04/11/2022	07/11/2022	08/11/2022	09/11/2022
		Tareas Pendientes			9	9	8	7	6	5	4	4	4	3	3	1	0
		Horas Pendientes			104	96	88	80	72	64	56	48	40	32	24	16	0

Backlog	Tarea	Categoría	Responsables	Estado	Estimación (Horas)	Esfuerzo
H.U. 01	Selección de artículos	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	16	8 8
H.U. 01	Extracción de modelos y/o algoritmos de Machine Learning	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	8	6

		Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X
		1	24/10/2022	8 horas	09/11/2022	24/10/2022	25/10/2022	26/10/2022	27/10/2022	28/10/2022	31/10/2022	01/11/2022	02/11/2022	03/11/2022	04/11/2022	07/11/2022	08/11/2022	09/11/2022
				Tareas Pendientes		9	9	8	7	6	5	4	4	4	3	3	1	0
				Horas Pendientes		104	96	88	80	72	64	56	48	40	32	24	16	0
H.U. 01	Extracción de valores del accuracy respectivamente	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	8			2	6									
H.U. 01	Documentación de la Frecuencia de los modelos y/o algoritmos de Machine Learning	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	8				2	4								

		Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X
		1	24/10/2022	8 horas	09/11/2022	24/10/2022	25/10/2022	26/10/2022	27/10/2022	28/10/2022	31/10/2022	01/11/2022	02/11/2022	03/11/2022	04/11/2022	07/11/2022	08/11/2022	09/11/2022
		Tareas Pendientes				9	9	8	7	6	5	4	4	4	3	3	1	0
		Horas Pendientes				104	96	88	80	72	64	56	48	40	32	24	16	0
H.U. 01	Selección de modelos y/o algoritmos de Machine Learning para realización de pruebas	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	8					4	2							
H.U. 01	Implementación de modelos y/o algoritmos de Machine Learning seleccionados	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	24						6	8	8	2				

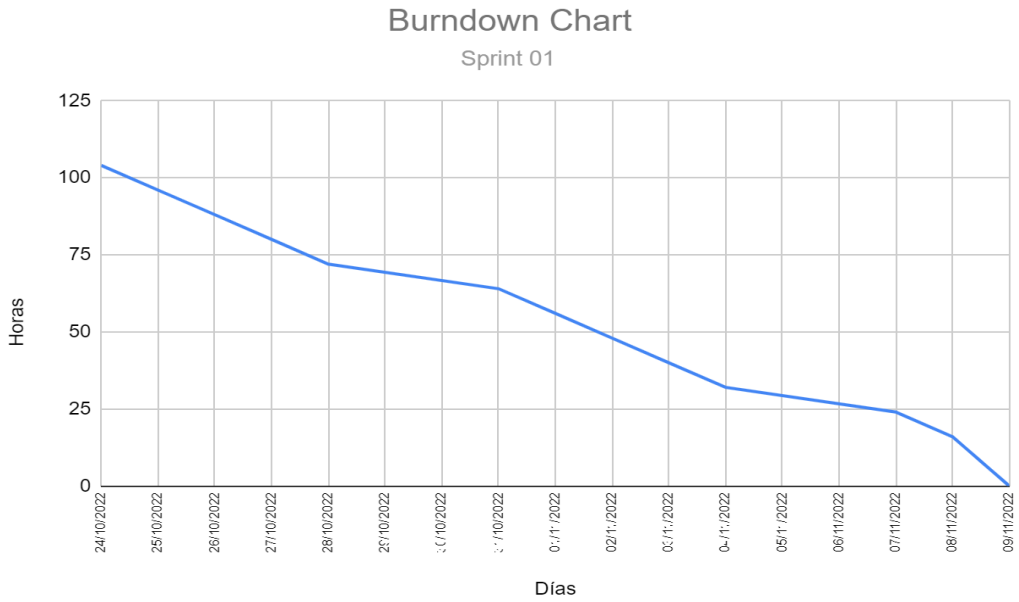
		Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X
		1	24/10/2022	8 horas	09/11/2022	24/10/2022	25/10/2022	26/10/2022	27/10/2022	28/10/2022	31/10/2022	01/11/2022	02/11/2022	03/11/2022	04/11/2022	07/11/2022	08/11/2022	09/11/2022
			Tareas Pendientes			9	9	8	7	6	5	4	4	4	3	3	1	0
			Horas Pendientes			104	96	88	80	72	64	56	48	40	32	24	16	0
H.U. 01	Ejecución de pruebas	Pruebas	Mishell Castillo y Kevin Chuquitarco	Finalizado	16									6	8	2		
H.U. 01	Documentación de métricas de evaluación resultantes	Documentación	Mishell Castillo y Kevin Chuquitarco	Finalizado	8											6		
H.U. 01	Selección de modelo y/o algoritmo de Machine Learning con mejor valor de accuracy	Documentación	Mishell Castillo y Kevin Chuquitarco	Finalizado	8												8	

Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X
1	24/10/2022	8 horas	09/11/2022	24/10/2022	25/10/2022	26/10/2022	27/10/2022	28/10/2022	31/10/2022	01/11/2022	02/11/2022	03/11/2022	04/11/2022	07/11/2022	08/11/2022	09/11/2022
Tareas Pendientes				9	9	8	7	6	5	4	4	4	3	3	1	0
Horas Pendientes				104	96	88	80	72	64	56	48	40	32	24	16	0
Esfuerzo Restante			104	96	88	80	72	64	56	48	40	32	24	16	8	8
Tendencia ideal			104	96	88	80	72	64	56	48	40	32	24	16	8	0

Burndown chart. Burndown chart o gráfico de avance es un diagrama que permite al equipo de desarrollo visualizar de forma efectiva el progreso del sprint, y con ello detectar posibles dificultades (Arroba Medina, 2011). En la Figura 7: Burndown Chart - Sprint 01, se muestra el avance realizado en el tiempo estimado para el desarrollo del presente sprint, en donde, el eje X muestra las fechas de los días especificados en la Tabla 9, en este caso el intervalo de tiempo inicia el 24/10/2022 y termina el 09/11/2022, en el eje Y en cambio se muestra el número total de horas estimadas al inicio, el cual se obtiene multiplicando el total de días estimado por las horas que se trabajará por día, que para este Sprint son 13 días y 8 horas diarias, lo que nos da un valor de 104 horas y que será el valor máximo de este eje, y que conforme avancen los días, el valor de horas debe ir disminuyendo con el objetivo de llegar a cero, completando el Sprint.

Figura 7

Burndown Chart - Sprint 01



Resultados del Sprint. En esta sección se explica brevemente el proceso realizado y los resultados más relevantes que se obtuvieron durante la ejecución del sprint y una vez finalizado el sprint. En el Anexo 1 se presentan los datos más relevantes sobre los artículos que se seleccionaron para realizar la revisión de la literatura, con la finalidad de tener un registro de estos y poder utilizarlo luego como base para la redacción del presente documento.

De acuerdo con la revisión de la literatura realizada y especificada en el capítulo II sección 1.2, se obtuvieron un total de 22 modelos y/o algoritmos de Machine Learning, se registró el valor máximo de precisión (accuracy) de cada modelo que permitía predecir si el o los sitios web tienen phishing en cada artículo en que se seleccionado. Además, se registró la frecuencia de cada uno, con los cuales se calculó el valor de la media de las frecuencias de cada modelo y/o algoritmo de Machine Learning extraído, y seleccionar aquellos que poseían una frecuencia igual o superior al valor de la media calculada. Finalmente, se elegio un total de 6 modelos y/o algoritmos de Machine Learning para ser probados en un futuro cercano, y se muestran en la Tabla 2 juntamente con una breve explicación.

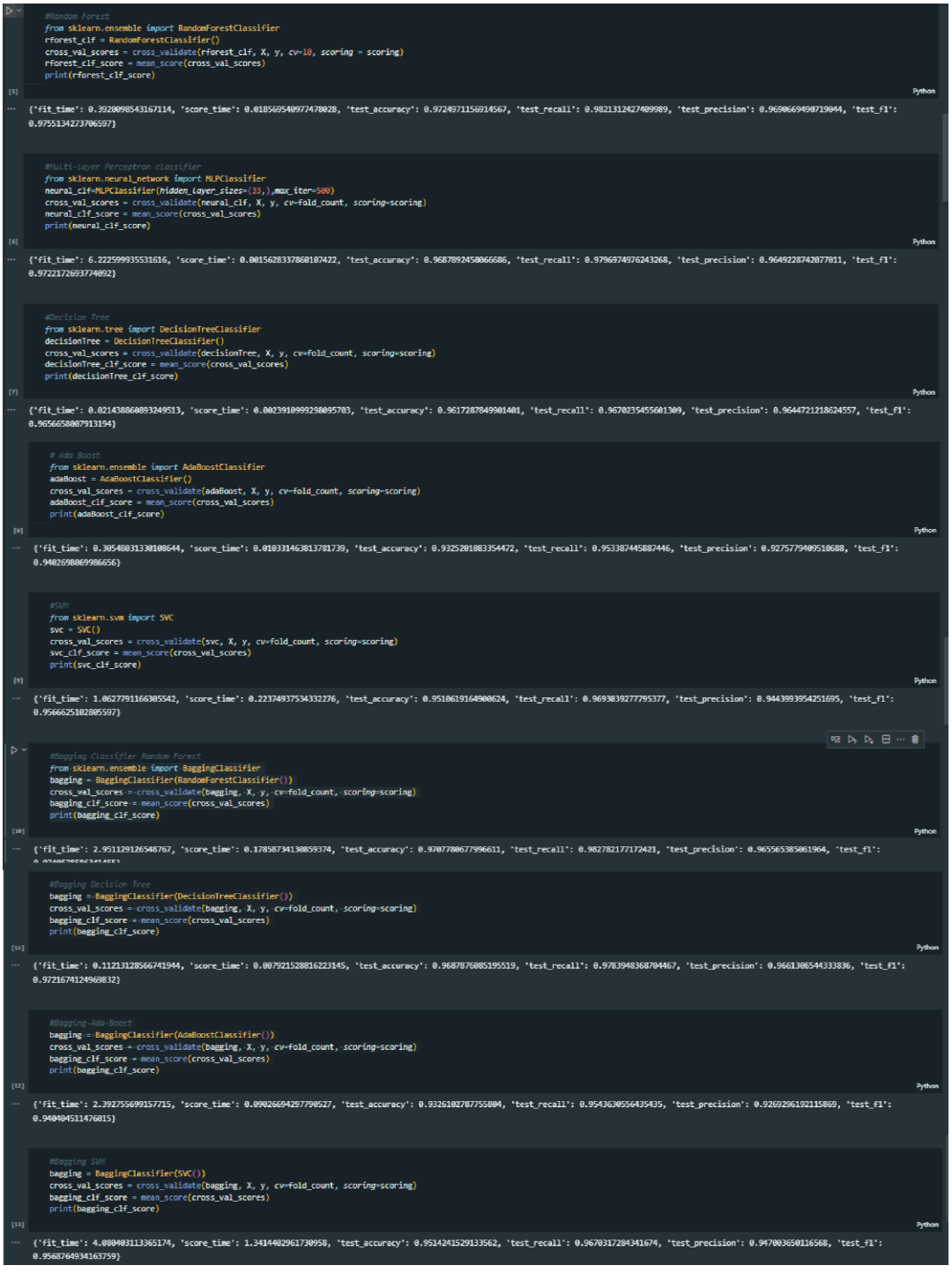
Una vez extraídos los modelos y/o algoritmos de Machine Learning que presentan los mejores valores de accuracy en la detección de phishing en sitios web, se procedió a implementarlos y probarlos utilizando el dataset que se encuentra en el estudio “Phishing Website Detection by Machine Learning Techniques” (Rashid et al., 2020), mismo que contiene URLs con 30 características extraídas. Fue necesario el uso del dataset mencionado porque el objetivo de este Sprint es encontrar el mejor modelo y/o algoritmo de Machine Learning. Cabe mencionar que en este Sprint aún no se cuenta con el dataset propuesto en este proyecto.

En la Figura 8: Implementación de modelos y/o algoritmos de Machine Learning, se puede evidenciar el código desarrollado para la implementación de los modelos y/o algoritmos de Machine Learning, con sus respectivas métricas de evaluación: accuracy, precision, recall y f1 score.

En la Tabla 10: Resultados, pruebas, modelos y/o algoritmos de Machine Learning implementados, se muestran las métricas de evaluación de los resultados obtenidos después de realizar las pruebas mencionadas a cada modelo y/o algoritmo seleccionado. Información que utilizamos para seleccionar el modelo que se va a implementar para el desarrollo del sistema, tomando como principal referencia los valores de accuracy, porque esta medida nos permite conocer el porcentaje de elementos correctamente clasificados entre positivos y negativos.

Figura 8

Implementación de modelos y/o algoritmos de Machine Learning



Como resultado de analizar los resultados mostrados en la Tabla 10, finalmente se seleccionó el modelo y/o algoritmo conocido como Random Forest, que presenta un valor de accuracy igual a 0,9725, que en términos porcentuales equivale a 97,25%.

Tabla 10

Resultados pruebas modelos y/o algoritmos de Machine Learning implementados

Características	Algoritmos/Modelos	Accuracy	Precision	Recall
30 características	Random Forest	0,9725	0,9691	0,9821
	Multi-layer Perceptron classifier	0,9688	0,9649	0,9797
	Decision Tree	0,9617	0,9645	0,9670
	Ada Boost	0,9325	0,9276	0,9534
	SVM	0,9511	0,9444	0,9693
	Bagging Random Forest	0,9708	0,9656	0,9828
	Bagging Decision Tree	0,9688	0,9661	0,9784
	Bagging Ada Boost	0,9326	0,9269	0,9544
	Bagging SVM	0,9514	0,9470	0,9569

Nota. Al finalizar este Sprint no se encontró mayor inconveniente para cumplir con el requerimiento establecido, debido a ello se decidió continuar utilizando la misma forma de trabajo.

Sprint 02: Creación del dataset

Para el desarrollo del presente Sprint, se tomó como base la Historia de Usuario H.U. 02 especificada en la Tabla 4, en primer lugar, se debe conocer y seleccionar las características que se pueden extraer de URLs, en segundo lugar, se crearán escenarios para

determinar las características que aportan en gran cantidad a la detección de sitios web con phishing, y finalmente se crea un dataset que sirva para entrenar al modelo seleccionado.

Historias de usuario detallada. La Tabla 11: Historia de usuario para la creación de un dataset, presenta la Historia de Usuario H.U. 02 del sistema de detección de phishing (Hunter Phisher) de forma detallada, donde se especifica los responsables del desarrollo, y los criterios de aceptación para la creación de un dataset, el cual contiene las características seleccionadas para el desarrollo del sistema propuesta.

Tabla 11

Historia de usuario para la creación de un dataset

Historias de Usuario	
Número: H.U. 02	**Usuario:** Usuario de internet
Nombre historia: Creación de dataset	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 20	**Interacción asignada:** 1
Programadores responsables: Mishell Castillo, Kevin Chuquitarco	
Descripción: ● Como usuario quiero un dataset que contenga características que permitan identificar sitios web con phishing de los legítimos.	
Validación (Criterios de aceptación): ● Se seleccionarán las características que tengan una frecuencia mayor o igual a la media de la frecuencia total. ● Se realizará pruebas con los modelos y/o algoritmos de Machine Learning implementados en diferentes escenarios, que difieren por la cantidad de características designadas para cada uno.	

Historias de Usuario	
Número: H.U. 02	**Usuario:** Usuario de internet
Nombre historia: Creación de dataset	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 20	**Interacción asignada:** 1
Programadores responsables: Mishell Castillo, Kevin Chuquitarco	

- Se realizará pruebas de los modelos y/o algoritmos de Machine Learning con la combinación de varios escenarios entre sí.
- Se extraerá las características seleccionadas a partir de una URL.
- El dataset se creará con las características de URL de sitios web legítimos y con phishing.

Sprint Backlog. En la Tabla 12: Sprint Backlog 02, se especifica las tareas que se realizaron para llevar a cabo el desarrollo del sprint, los responsables de realizar cada uno de estos, las respectivas fechas en las que se planificó ejecutar el sprint, el tiempo que se estimó en horas, el esfuerzo en horas que se trabajó realmente cada día y el estado en el que actualmente se encuentra cada tarea. Cabe mencionar que se presenta el sprint backlog ya finalizado.

Tabla 12

Sprint Backlog 02

	Sprint	Inicio	Jornada	Fin	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X
	2	10/11/2022	8 horas	07/12/2022	10/11/2022	11/11/2022	14/11/2022	15/11/2022	16/11/2022	17/11/2022	18/11/2022	21/11/2022	22/11/2022	23/11/2022	24/11/2022	25/11/2022	28/11/2022	29/11/2022	30/11/2022	01/12/2022	02/12/2022	05/12/2022	06/12/2022	07/12/2022	
	Tareas Pendientes				13	13	12	11	8	8	7	6	5	3	2	2	2	1	1	0	0	0	0	0
	Horas Pendientes				160	152	144	135	119	103	87	71	55	47	39	31	23	15	7	0	0	0	0	0

Backlog	Tarea	Categoría	Responsables	Estado	Estimación (Horas)	Esfuerzo
H.U. 02	Extracción de características de URL para identificar sitios web con phishing	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	16	8 8
H.U. 02	Contabilización de frecuencias de cada característica extraída	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	6	6

		Sprint	Inicio	Jornada	Fin	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X
		2	10/11/2022	8 horas	07/12/2022	10/11/2022	11/11/2022	14/11/2022	15/11/2022	16/11/2022	17/11/2022	18/11/2022	21/11/2022	22/11/2022	23/11/2022	24/11/2022	25/11/2022	28/11/2022	29/11/2022	30/11/2022	01/12/2022	02/12/2022	05/12/2022	06/12/2022	07/12/2022
		Tareas Pendientes				13	13	12	11	8	8	7	6	5	3	2	2	2	1	1	0	0	0	0	0
		Horas Pendientes				160	152	144	135	119	103	87	71	55	47	39	31	23	15	7	0	0	0	0	0
H.U. 02	Selección de características para identificar sitios web con phishing para realizar pruebas	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	6			2	4																
H.U. 02	Ordenación de las características por relevancia	Revisión de literatura	Mishell Castillo y Kevin Chuquitarco	Finalizado	1				1																
H.U. 02	Creación de escenarios con diferentes cantidades de características para identificar sitios web con phishing	Documentación	Mishell Castillo y Kevin Chuquitarco	Finalizado	2				2																

		Sprint	Inicio	Jornada	Fin	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X
		2	10/11/2022	8 horas	07/12/2022	10/11/2022	11/11/2022	14/11/2022	15/11/2022	16/11/2022	17/11/2022	18/11/2022	21/11/2022	22/11/2022	23/11/2022	24/11/2022	25/11/2022	28/11/2022	29/11/2022	30/11/2022	01/12/2022	02/12/2022	05/12/2022	06/12/2022	07/12/2022
		Tareas Pendientes				13	13	12	11	8	8	7	6	5	3	2	2	2	1	1	0	0	0	0	0
		Horas Pendientes				160	152	144	135	119	103	87	71	55	47	39	31	23	15	7	0	0	0	0	0
H.U. 02	Implementación de los escenarios de características con los modelos y/o algoritmos de Machine Learning	Codificación	Kevin Chuquitarco	Finalizado	14				1	8	5														
H.U. 02	Ejecución de pruebas con los escenarios creados	Codificación	Kevin Chuquitarco	Finalizado	11						3	8													
H.U. 02	Documentación de métricas de evaluación resultantes	Documentación	Kevin Chuquitarco	Finalizado	8								8												

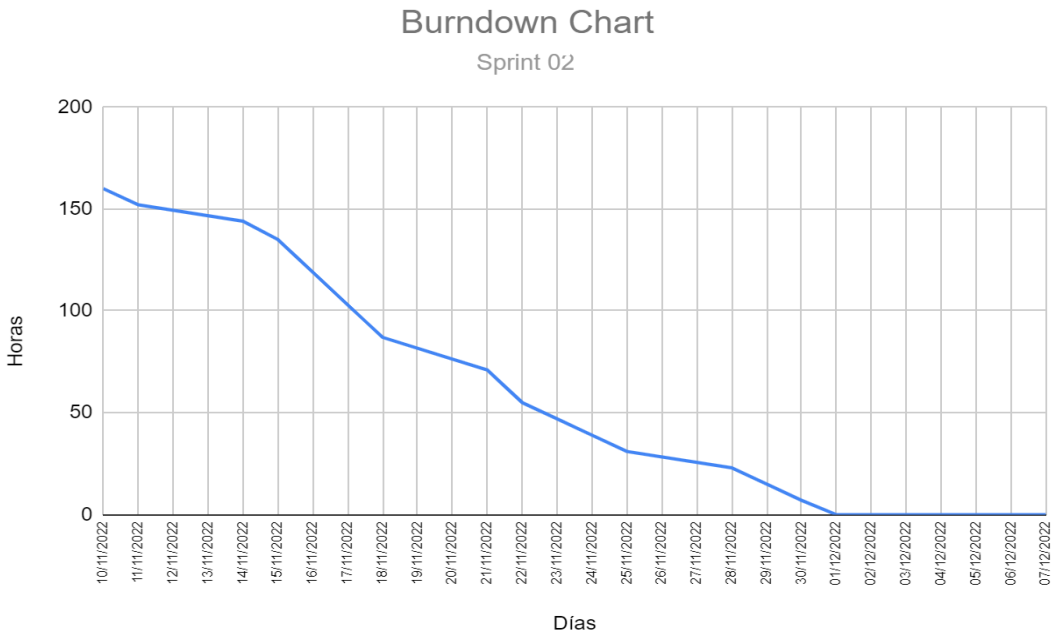
		Sprint	Inicio	Jornada	Fin	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X
		2	10/11/2022	8 horas	07/12/2022	10/11/2022	11/11/2022	14/11/2022	15/11/2022	16/11/2022	17/11/2022	18/11/2022	21/11/2022	22/11/2022	23/11/2022	24/11/2022	25/11/2022	28/11/2022	29/11/2022	30/11/2022	01/12/2022	02/12/2022	05/12/2022	06/12/2022	07/12/2022
		Tareas Pendientes				13	13	12	11	8	8	7	6	5	3	2	2	2	1	1	0	0	0	0	0
		Horas Pendientes				160	152	144	135	119	103	87	71	55	47	39	31	23	15	7	0	0	0	0	0
H.U. 02	Selección de características significativas para la detección de sitios web con phishing	Documentación	Kevin Chuquitarco	Finalizado	8									8											
H.U. 02	Selección de datasets con sitios web legítimos y con phishing	Investigación	Mishell Castillo y Kevin Chuquitarco	Finalizado	8										8										
H.U. 02	Limpieza y unión de datasets seleccionados	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	20											8	8	4							
H.U. 02	Implementación de código para la extracción de características de una URL	Codificación	Mishell Castillo	Finalizado	41				1	8	8	8	8	8											

		Sprint	Inicio	Jornada	Fin	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X
		2	10/11/2022	8 horas	07/12/2022	10/11/2022	11/11/2022	14/11/2022	15/11/2022	16/11/2022	17/11/2022	18/11/2022	21/11/2022	22/11/2022	23/11/2022	24/11/2022	25/11/2022	28/11/2022	29/11/2022	30/11/2022	01/12/2022	02/12/2022	05/12/2022	06/12/2022	07/12/2022
		Tareas Pendientes				13	13	12	11	8	8	7	6	5	3	2	2	2	1	1	0	0	0	0	0
		Horas Pendientes				160	152	144	135	119	103	87	71	55	47	39	31	23	15	7	0	0	0	0	0
H.U. 02	Implementación de código para la creación de dataset	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	19													4	8	7					
	Esfuerzo Restante				16 0	15	14	13	12	11	95	79	63	47	39	31	23	15	7	0	0	0	0	0	0
	Tendencia ideal				16 0	15	14	13	12	12	11	10	96	88	80	72	64	56	48	40	32	24	16	8	0

Burndown chart. En la Figura 9: Burndown Chart - Sprint 02, se muestra el avance realizado en el tiempo estimado para el desarrollo del presente sprint, donde en el eje X se muestran las fechas de los días especificados en la Tabla 9, en este caso el intervalo de tiempo inicia el 24/10/2022 y termina el 09/11/2022, en el eje Y en cambio se muestra el número total de horas estimadas al inicio, el cual se obtiene multiplicando el total de días estimado por las horas que se trabajará por día, que para este Sprint son 13 días y 8 horas diarias, lo que nos da un valor de 104 horas y que será el valor máximo de este eje, y que conforme avancen los días, el valor de horas debe ir disminuyendo con el objetivo de llegar a cero, completando el Sprint.

Figura 9

Burndown Chart - Sprint 02



Resultados del Sprint. En esta sección se explican brevemente el proceso realizado y los resultados más relevantes que se obtuvieron durante la ejecución del Sprint 02 y una vez finalizado el sprint. En el Anexo 3 se presentan todas las características (recursos de comprobación) extraídas de los artículos presentados en el Anexo 1 y de los dataset seleccionados del repositorio KAGGLE, cuya información primordial se presenta en el Anexo 2; además de una breve descripción se presentan cada una de las características.

Se obtuvieron un total de 62 recursos de comprobación, seguido se procedió a registrar la frecuencia con la que aparecía, para después calcular el valor de la media de las frecuencias de cada recurso de comprobación extraído, se seleccionaron aquellos que poseían una frecuencia igual o superior al valor de la media calculada, con este procedimiento se elegio un total de 30 características para ser probados en un futuro cercano, y se muestra cada una de ellas con su respectiva descripción en la Tabla 1. Estas características se ordenaron descendentemente de acuerdo a su valor de incidencia, como la relevancia de esta, tal como se muestra en la Tabla 13: Características ordenadas por relevancia.

Tabla 13

Características ordenadas por relevancia

ORD.	ID	CARACTERÍSTICAS	INCIDENCIA
1	1	Tiene Dirección IP	21
2	2	Longitud del URL	21
3	4	Tiene el símbolo @	21
4	8	Estado SSL (Secure Sockets Layer)	20
5	24	Edad del Dominio	20
6	5	Redirección de doble barra	19
7	14	URL de anclaje	19
8	6	Prefijo Sufijo	18

ORD.	ID	CARACTERÍSTICAS	INCIDENCIA
9	15	Enlaces en Etiquetas	17
10	21	Deshabilitar clic derecho	17
11	22	Uso de la ventana emergente	17
12	10	Favicon	16
13	18	URL anormal	16
14	23	Iframe	16
15	25	Registro DNS	16
16	28	Índice de google	16
17	11	Puerto	15
18	13	Request URL	15
19	16	Controlador de formulario de servidor (SFH)	15
20	19	Recuento de redirección del sitio web (website forwarding)	15
21	20	Personalización de la barra de estado (Mouse over)	15
22	26	Tráfico web	15
23	3	Servicio de acortamiento	14
24	9	Duración del registro del dominio	14
25	12	Token HTTPS	14
26	17	Envío de información al correo electrónico	14
27	27	Rango de página	14
28	30	Informe estadístico	14
29	7	Tiene un subdominio	13
30	29	Enlaces que apuntan a la página	12

Además, en este Sprint se implementa el código para probar diferentes escenarios establecidos para validación, para ello, en cada escenario se conforman grupos de 10 características (recursos de comprobación). La Tabla 13 muestra las características, encontradas en los artículos seleccionados en la revisión de literatura realizada, para detectar phishing de acuerdo con su incidencia (valor medio) y relevancia (ordenadas). Se crean 3 escenarios, la razón de estos es probar cual es el aporte de las características más relevantes, las menos relevantes y las que están en el medio, así como de la combinación de estos grupos, teniendo como objetivo de determinar las características que más aporten a la detección de phishing en sitios web, para lo cual los modelos se probaron de forma individual y combinada entre ellos, por ejemplo: en primer lugar se probó las diez primeras características, luego se probaron las segundas diez, para después probar las diez restantes, luego se probaron las diez primeras conjuntamente con las diez segundas, después se probaron las diez primeras y las diez terceras características, y finalmente se probaron las diez segundas y diez terceras, como se puede evidenciar en la Figura 10: Pruebas de características con diferentes escenarios. El resultado de las treinta características unidas fue probado en el Sprint 01, específicamente en la Tabla 10. Es importante mencionar que las pruebas se realizaron utilizando los 6 modelos y/o algoritmos de Machine Learning seleccionados y mostrados en la Tabla 2.

En la Tabla 14: Resultados pruebas modelos y/o algoritmos de Machine Learning implementados con los diferentes escenarios, se presenta el resultado de la ejecución del código que se muestra en la Figura 10, los mismos que se sometieron a las mismas métricas de evaluación especificadas al inicio de este capítulo, es decir, para medir el rendimiento de los algoritmos de Machine Learning se propuso tres métricas: Acurracy, Precisión y Recall. Para la ejecución de estas pruebas se utilizó el dataset que se encuentra en el estudio “Phishing Website Detection by Machine Learning Techniques” (Rashid et al., 2020), que contenía diversidad de URLs legítimas, así como también, URLs con phishing.

En primer lugar, se aplicó la métrica exactitud (accuracy), la cual determina el porcentaje de los sitios web con phishing y legítimos bien clasificados con respecto a todos los datos de entrenamiento, razón por la cual se consideró como la métrica más importante de evaluación, en el primer escenario (con las diez primeras características) se obtuvo como mejor accuracy el valor de 0,9318 (93,18%) proveniente del algoritmo Random Forest. En el segundo escenario (las segundas diez características) se tuvo como mejor accuracy 0,6943 (69,43%) proveniente del algoritmo Random Forest combinado con el algoritmo Bagging. Finalmente, en el tercer escenario (con las terceras diez características) se obtuvo como mejor accuracy 0,7953 (79,53%) proveniente del algoritmo Random Forest. Analizando los resultados obtenidos en los tres escenarios probados de manera independiente, se puede determinar que las primeras 10 características son las que más aportan en la detección de phishing en sitios web. Por otro lado, se combinó estos escenarios, con la finalidad de conocer cuál de estas combinaciones aumentan y/o disminuyen el valor de accuracy de detección. Se inició con el primer y segundo escenario combinados (diez primeras juntamente con las diez segundas características), en donde, se obtuvo como mejor accuracy 0.9523 (95,23%) proveniente del algoritmo Random Forest. Después se combinó el primer y tercer escenario (diez primeras juntamente con las diez terceras características) el mejor accuracy fue de 0.9638 (96,38%) proveniente del algoritmo Random Forest y seguido, se combinó el segundo y tercer escenario (diez segundas juntamente con las diez terceras características) para obtener como mejor accuracy el valor de 0,8562 (85,62%) proveniente del algoritmo Random Forest. Finalmente se combinaron los tres escenarios (treinta características totales) para obtener como mejor accuracy el valor de 0,9725 (97,25%) proveniente del algoritmo Random Forest. Es importante mencionar que, los resultados de las treinta características se encuentran en la Tabla 10.

La métrica precisión (precision) determina el porcentaje de sitios web con phishing bien clasificados con respecto a todos los sitios web clasificados como phishing (sitios con phishing clasificados como phishing y sitios legítimos clasificados como phishing), además se realizó el

mismo proceso que en la métrica accuracy, teniendo en el primer escenario (con las diez primeras características) la mejor precisión 0,9209 (92,09%) proveniente del algoritmo Decision-Tree. En el segundo escenario (las segundas diez características) que tuvo como mejor Precisión 0,7057 (70,57%) proveniente del algoritmo Decision-Tree. Finalmente, en el tercer escenario (con las terceras diez características) se obtuvo como mejor Precisión 0,7987 (79,87%) proveniente del algoritmo Decision-Tree. Los resultados obtenidos en los tres escenarios probados de manera independiente, nos indica que las primeras 10 características son las que más aportan en la detección de phishing en sitios web. De igual manera, se combinó estos escenarios, con la finalidad de conocer cuál de estas combinaciones aumentan y/o disminuyen la precisión de detección. Se inició con el primer y segundo escenario combinados (diez primeras juntamente con las diez segundas características), en donde, se obtuvo como mejor Precisión un 0.9513 (95,13%) proveniente del algoritmo Random Forest. Después se combinó el primer y tercer escenario (diez primeras juntamente con las diez terceras características) la mejor Precisión fue de 0.9626 (96,26%) proveniente del algoritmo Random Forest y se combinó el segundo y tercer escenario (diez segundas juntamente con las diez terceras características) para obtener como mejor Precisión el valor de 0,8577 (85,77%) proveniente del algoritmo Random Forest. Finalmente se combinaron los tres escenarios (treinta características totales) para obtener como mejor precisión 0,9691 (96,91%) proveniente del algoritmo Random Forest. Con esta métrica de evaluación se determina que el algoritmo Decision-Tree presenta buenos resultados con menor cantidad de características, por otro lado, Random Forest aumenta su Precisión con mayor cantidad de características, así como también se evidencia que los mejores resultados se obtienen utilizando todas las características, es decir las 30 características con el algoritmo Random Forest.

Para la métrica Recall, la cual determina el porcentaje de sitios web con phishing bien clasificados con respecto a todas las instancias de entrenamiento de sitios web con phishing, y se realizó el mismo proceso que en la métrica accuracy y Precisión, teniendo en el primer

escenario (con las diez primeras características) el mejor Recall 0,9605 (96,05%) proveniente del algoritmo Random Forest. En el segundo escenario (las segundas diez características) se tuvo como mejor Recall 0,7790 (77,90%) proveniente del algoritmo Random Forest, combinado con el algoritmo Bagging. Finalmente, en el tercer escenario (con las terceras diez características) se obtuvo un Recall 0,8559 (85,59%) proveniente del algoritmo Random Forest. De igual manera, los resultados obtenidos en los tres escenarios probados de manera independiente, nos indica que las primeras 10 características son las que más aportan en la detección de phishing en sitios web. Se combinaron estos escenarios, con la finalidad de conocer cuál de estas combinaciones aumentan y/o disminuyen la métrica de evaluación Recall. Se inició con el primer y segundo escenario combinados (diez primeras juntamente con las diez segundas características), en donde, se obtuvo como mejor Recall 0.9670 (96,70%) proveniente del algoritmo Support Vector Machine (SVM). Después se combinó el primer y tercer escenario (diez primeras juntamente con las diez terceras características) el mejor Recall fue de 0.9730 (97,30%) proveniente del algoritmo Random Forest y seguido, se combinó el segundo y tercer escenario (diez segundas juntamente con las diez terceras características) para obtener como mejor Recall el valor de 0,8947 (89,47%) proveniente del algoritmo Random Forest. Finalmente, se combinaron los tres escenarios (treinta características totales) para obtener como mejor Recall el valor de 0,9828 (98,28%) proveniente del algoritmo Random Forest combinado con el algoritmo Bagging. Con esta métrica de evaluación se observa que los mejores resultados se obtienen utilizando todas las treinta características con el algoritmo Random Forest combinado con el algoritmo Bagging.

Figura 10

Pruebas de características con diferentes escenarios

Tabla 14

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

Ord.	Características	Algoritmos / modelos	Accuracy	Precision	Recall
1	Primeras 10 Características	Random Forest	0,9318	0,9208	0,9605
		Multi-layer Perceptron classifier	0,9289	0,9183	0,9581
		Decision Tree	0,9304	0,9209	0,9576
		Ada Boost	0,9217	0,9161	0,9462
		SVM	0,9281	0,9154	0,9599

Ord.	Características	Algoritmos / modelos	Accuracy	Precision	Recall
		Bagging Random Forest	0,9304	0,9189	0,9599
		Bagging Decision Tree	0,9304	0,9196	0,9592
		Bagging Ada Boost	0,9217	0,9156	0,9469
		Bagging SVM	0,9279	0,9160	0,9587
2	Segundas 10 Características	Random Forest	0,6937	0,7037	0,7788
		Multi-layer Perceptron classifier	0,6908	0,7028	0,7715
		Decision Tree	0,6937	0,7057	0,7735
		Ada Boost	0,6620	0,6743	0,7603
		SVM	0,6908	0,7028	0,7715
		Bagging Random Forest	0,6943	0,7041	0,7790
		Bagging Decision Tree	0,6909	0,7008	0,7780
		Bagging Ada Boost	0,6623	0,6748	0,7597
		Bagging SVM	0,6921	0,7025	0,7762
3	Terceras 10 Características	Random Forest	0,7953	0,7972	0,8559
		Multi-layer Perceptron classifier	0,7774	0,7813	0,8425
		Decision Tree	0,7927	0,7987	0,8470
		Ada Boost	0,7538	0,7628	0,8224
		SVM	0,7773	0,7801	0,8454
		Bagging Random Forest	0,7933	0,7960	0,8532
		Bagging Decision Tree	0,7923	0,7967	0,8490
		Bagging Ada Boost	0,7546	0,7546	0,8219
		Bagging SVM	0,7808	0,7847	0,8439

Ord.	Características	Algoritmos / modelos	Accuracy	Precision	Recall
4	Primeras 10 y Segundas 10 Características	Random Forest	0,9523	0,9513	0,9639
		Multi-layer Perceptron classifier	0,9494	0,9475	0,9626
		Decision Tree	0,9487	0,9483	0,9604
		Ada Boost	0,9273	0,9169	0,9563
		SVM	0,9396	0,9277	0,9670
		Bagging Random Forest	0,9500	0,9470	0,9643
		Bagging Decision Tree	0,9503	0,9484	0,9633
		Bagging Ada Boost	0,9266	0,9161	0,9558
		Bagging SVM	0,9410	0,9303	0,9667
5	Primeras 10 y Terceras 10 Características	Random Forest	0,9638	0,9626	0,9730
		Multi-layer Perceptron classifier	0,9578	0,9589	0,9659
		Decision Tree	0,9560	0,9613	0,9599
		Ada Boost	0,9266	0,9227	0,9479
		SVM	0,9441	0,9378	0,9638
		Bagging Random Forest	0,9624	0,9611	0,9719
		Bagging Decision Tree	0,9615	0,9621	0,9691
		Bagging Ada Boost	0,9265	0,9222	0,9482
		Bagging SVM	0,9431	0,9386	0,9609
6	Segundas 10 y las Terceras 10 Características	Random Forest	0,8562	0,8577	0,8947
		Multi-layer Perceptron classifier	0,8394	0,8438	0,8774
		Decision Tree	0,8459	0,8529	0,8790

Ord.	Características	Algoritmos / modelos	Accuracy	Precision	Recall
		Ada Boost	0,7611	0,7772	0,8142
		SVM	0,8304	0,8318	0,8803
		Bagging Random Forest	0,8511	0,8513	0,8933
		Bagging Decision Tree	0,8496	0,8531	0,8870
		Bagging Ada Boost	0,7615	0,7772	0,8160
		Bagging SVM	0,8271	0,8365	0,8660

En la Tabla 15: Ganador de cada escenario, se muestra los modelos y/o algoritmos de Machine Learning que obtuvieron la mejor evaluación en cada métrica aplicada en cada escenario. Para la métrica Accuracy se evidencia que en todos los escenarios propuestos el algoritmo Random Forest, ya sea combinado o sin combinar con el algoritmo de optimización Bagging, se obtiene el mayor porcentaje de detección de Sitios Web con phishing y legítimos bien clasificados, con respecto a todos los datos de entrenamiento. En cuanto a la métrica Precision, se puede visualizar que el algoritmo Decision Tree presenta los mejores resultados en los tres primeros escenarios (en donde, se tiene solamente 10 características), mientras que, en los otros escenarios el algoritmo Random Forest es el que predomina. En cuanto a la métrica Recall se observa nuevamente que el algoritmo Random Forest, combinado o sin combinar con el algoritmo optimización Bagging, se obtiene el mayor porcentaje de detección de Sitios Web con phishing y legítimos, teniendo como ganador al algoritmo SVM en una sola incidencia, en el Primer y Segundo Escenario (Primeras 10 y Segundas 10 Características). De los resultados obtenidos, en los distintos escenarios, se puede indicar que el escenario denominado: “primer escenario combinado con el tercer escenario” tiene la más alto Accuracy (Exactitud) al detectar el ataque phishing para sitios web.

De acuerdo, a la revisión realizada se encontró que el porcentaje de Accuracy más alto es del 99,57% obtenida en el artículo de (Somesha et al., 2020) y el valor más bajo es de 91,46% proveniente es del estudio de (Chapla et al., 2019). Con las pruebas realizadas al modelo propuesto en este trabajo se obtuvo un 97,25% de Accuracy más alto y un Accuracy más bajo del 93,25% usando las 30 características seleccionadas, con estos valores se puede indicar que se obtuvieron porcentajes dentro de los rangos establecidos en 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.

Tabla 15

Ganador de cada escenario

ORD.	ESCENARIO / MÉTRICA	ACCURACY	PRECISION	RECALL
1	Primer Escenario (Primeras 10 Características)	Random Forest (93,18%)	Decision-Tree (92,09%)	Random Forest (96,05%)
2	Segundo Escenario (Segundas 10 Características)	Bagging Random Forest (69,43%)	Decision-Tree (70,57%)	Bagging Random Forest (77,90%)
3	Tercer Escenario (Terceras 10 Características)	Random Forest (79,53%)	Decision-Tree (79,87%)	Random Forest (85,59%)
4	Primer y Segundo Escenario (Primeras 10 y Segundas 10 Características)	Random Forest (95,23%)	Random Forest (95,13%)	SVM (96,70%)

ORD.	ESCENARIO / MÉTRICA	ACCURACY	PRECISION	RECALL
5	Primer y Tercer Escenario (Primeras 10 y Terceras 10 Características)	Random Forest (96,38%)	Random Forest (96,26%)	Random Forest (97,30%)
6	Segundo y Tercer Escenario (Segundas 10 y las Terceras 10 Características)	Random Forest (85,62%)	Random Forest (85,77%)	Random Forest (89,47%)
7	Primer, Segundo y Tercer Escenario (30 Características)	Random Forest (97,25%)	Random Forest (96,91%)	Bagging Random Forest (98,28%)

Creación del dataset

Para la creación del dataset solicitado en la H.U. 02, se procedió a desarrollar el código correspondiente para extraer las 30 características seleccionadas a partir de una URL, el cuál es primordial para la creación del dataset. Cada característica fue implementada en un método dentro de una clase, la cual retorna 1, 0 o -1, donde 1 es legítimo, 0 es sospechoso y -1 es phishing. En la Figura 11: Extracción de características del sitio web YouTube, se muestra la ejecución del código para extraer las características a partir de una URL, el mismo que muestra la URL enviada (https://www.youtube.com), la respuesta del sitio web (<Response [200]>) y un arreglo que contiene las 30 características obtenidas de la URL enviada.

Figura 11

Extracción de características del sitio web Youtube

```
In [1]: runfile('C:/Users/Mishell/Desktop/TITULACION/ids-phishing/crear
dataset/feature_extraction.py', wdir='C:/Users/Mishell/Desktop/
TITULACION/ids-phishing/crear dataset')
La url ingresada es https://www.youtube.com/
<Response [200]>
[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, 0, 1]
```

Una vez desarrollado, probado y ejecutado el código para extraer características, se seleccionó el dataset denominado como Phishing Websites Dataset (Ariyadasa et al., 2021) almacenado en el repositorio Mendeley Data, el cual contiene instancias de sitios web legítimos y sitios web con phishing etiquetados respectivamente. Se desarrollo un código para “limpiar” el dataset mencionado, que principalmente se encarga de leer el dataset y enviar cada URL a realizar un servicio GET con un límite de tiempo de 5 seg, para después en base sobre el resultado que retorne, poder filtrar las URLs y seleccionar sólo las que dan una respuesta equivalente a 200, respuestas con otro valor fueron descartadas, una respuesta 200 representa que la solicitud realizada tuvo éxito.

Inicialmente el dataset seleccionado contenía un total de 42.000 URLs, entre los cuales había 30.000 URLs con phishing y 12.000 URLs legítimas, que después de ejecutar el código mostrado, se obtuvo como resultado un total de 11.743 URLs que retornaban una respuesta exitosa, entre los cuales había 2.546 URLs con phishing y 9.197 URLs legítimas que fueron guardados en un archivo CSV.

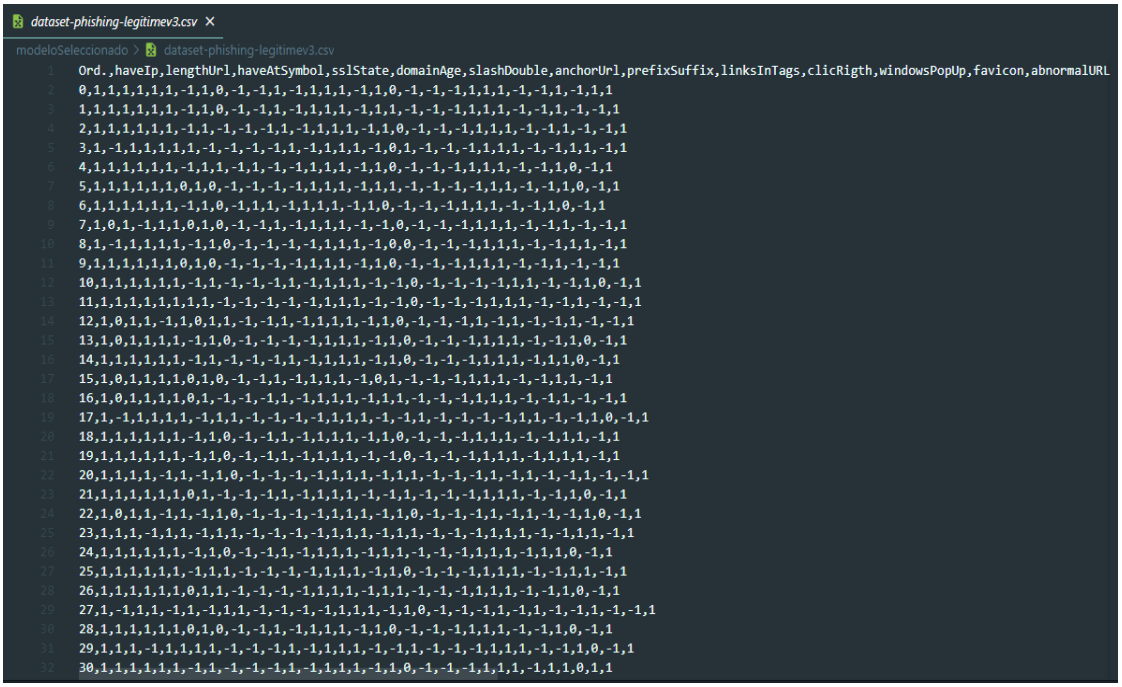
Una vez obtenido el archivo CSV con los 2.546 URLs con phishing y 9.197 URLs legítimas, se ejecutó el código dedicado específicamente para la creación del dataset. Este código se encarga de leer el archivo CSV creado anteriormente, enviar ordenadamente cada URL a extraer las características correspondientes y almacenarlos en una lista local, para

después almacenarlos en un archivo CSV, creando finalmente el dataset que se utilizará para entrenar el modelo implementado. En la Figura 12: Dataset creado, se muestra una captura de pantalla con una pequeña parte del dataset creado en este sprint, que contiene solamente las características extraídas de cada URL de manera ordenada con su respectivo resultado que puede ser entre: sitio web legítimo (1) o sitio web con phishing (-1).

Nota. El mayor problema identificado en este Sprint se relaciona directamente con los recursos computacionales al momento de ejecutar el código para crear el dataset, razón por la cual, se organizó para utilizar el computador más potente entre los integrantes del proyecto para cualquier ejecución final de algún código que se desarrolle a partir del presente Sprint, así poder avanzar y cumplir con los tiempos estimados en el Sprint Backlog.

Figura 12

Dataset creado



Sprint 03: Creación de la API

Para el desarrollo del presente Sprint, se tomó como base la Historia de Usuario H.U. 03 especificada en la Tabla 4, en primer lugar, se debe entrenar el modelo y/o algoritmo de

Machine Learning con el dataset creado, en segundo lugar, guardar el modelo entrenado, y finalmente se creará una API con las rutas solicitadas para realizar predicciones a través de una URL.

Historias de usuario detalladas. La Tabla 16: Historia de usuario para la creación de la API, presenta la Historia de Usuario H.U. 03 del sistema de detección de phishing (Hunter Phisher) de forma detallada, en donde se especifica los responsables del desarrollo, y los criterios de aceptación para la creación de la API.

Tabla 16

Historia de usuario para la creación de la API

Historias de Usuario	
Número: H.U. 01	**Usuario:** Usuario de internet
Nombre historia: Creación de API	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 20	**Interacción asignada:** 1
Programadores responsables: Mishell Castillo, Kevin Chuquitarco	
Descripción: ● Como usuario quiero que el modelo de ML se encuentre almacenado en un servidor y pueda realizar predicciones a través de un servicio.	
Validación (Criterios de aceptación): ● La API se almacenará en un servidor. ● Se implementarán dos rutas, la primera para realizar la predicción donde será obligatorio el atributo URL, la segunda será la ruta raíz donde se mostrará un mensaje de bienvenida. ● Se extraerán las características del atributo URL enviado a través de la primera ruta.	

Historias de Usuario	
Número: H.U. 01	**Usuario:** Usuario de internet
Nombre historia: Creación de API	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 20	**Interacción asignada:** 1
Programadores responsables: Mishell Castillo, Kevin Chuquitarco	

- La API deberá realizar la predicción utilizando el modelo que estará almacenado en el mismo servidor.
- La respuesta de la API será 1 cuando el sitio web es legítimo y -1 cuándo tiene phishing.

Sprint Backlog. En la Tabla 17: Sprint Backlog 03, se especifica las tareas que se realizaron para llevar a cabo el desarrollo del Sprint, los responsables de realizar cada uno de estas, las respectivas fechas en las que se planificó ejecutar el Sprint, el tiempo que se estimó en horas, el esfuerzo en horas que se trabajó realmente cada día y el estado en el que actualmente se encuentra cada tarea. Cabe mencionar que se presenta el Sprint Backlog ya finalizado.

Tabla 17

Sprint Backlog 03

Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
3	05/12/2022	8 horas	30/12/2022	05/12/2022	06/12/2022	07/12/2022	08/12/2022	09/12/2022	12/12/2022	13/12/2022	14/12/2022	15/12/2022	16/12/2022	19/12/2022	20/12/2022	21/12/2022	22/12/2022	23/12/2022	26/12/2022	27/12/2022	28/12/2022	29/12/2022	30/12/2022
Tareas Pendientes				12	9	9	9	9	9	9	9	9	8	8	7	7	5	5	3	3	1	1	0
Horas Pendientes				160	144	136	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0

Backlog	Tarea	Categoría	Responsables	Estado	Estimación (Horas)	Esfuerzo
H.U. 03	Guardar el modelo de Machine Learning entrenado	Codificación	Mishell Castillo	Finalizado	8	8
H.U. 03	Preparación del entorno local	Codificación	Kevin Chuquitarco	Finalizado	4	4
H.U. 03	Desarrollar API	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	60	8 8 8 8 8 8 8 4

		Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
		3	05/12/2022	8 horas	30/12/2022	05/12/2022	06/12/2022	07/12/2022	08/12/2022	09/12/2022	12/12/2022	13/12/2022	14/12/2022	15/12/2022	16/12/2022	19/12/2022	20/12/2022	21/12/2022	22/12/2022	23/12/2022	26/12/2022	27/12/2022	28/12/2022	29/12/2022	30/12/2022
			Tareas Pendientes			12	9	9	9	9	9	9	9	9	8	8	7	7	5	5	3	3	1	1	0
			Horas Pendientes			160	144	136	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
H.U. 03	Pruebas de API	Pruebas	Mishell Castillo y Kevin Chuquitarco	Finalizado	16									4	8	4									
H.U. 03	Búsqueda de servidores	Investigación	Kevin Chuquitarco	Finalizado	4	4																			
H.U. 03	Cotización y selección de un servidor	Investigación	Mishell Castillo y Kevin Chuquitarco	Finalizado	16											4	8	4							
H.U. 03	Contratación de un plan básico del servidor seleccionado	Investigación	Mishell Castillo y Kevin Chuquitarco	Finalizado	4													4							

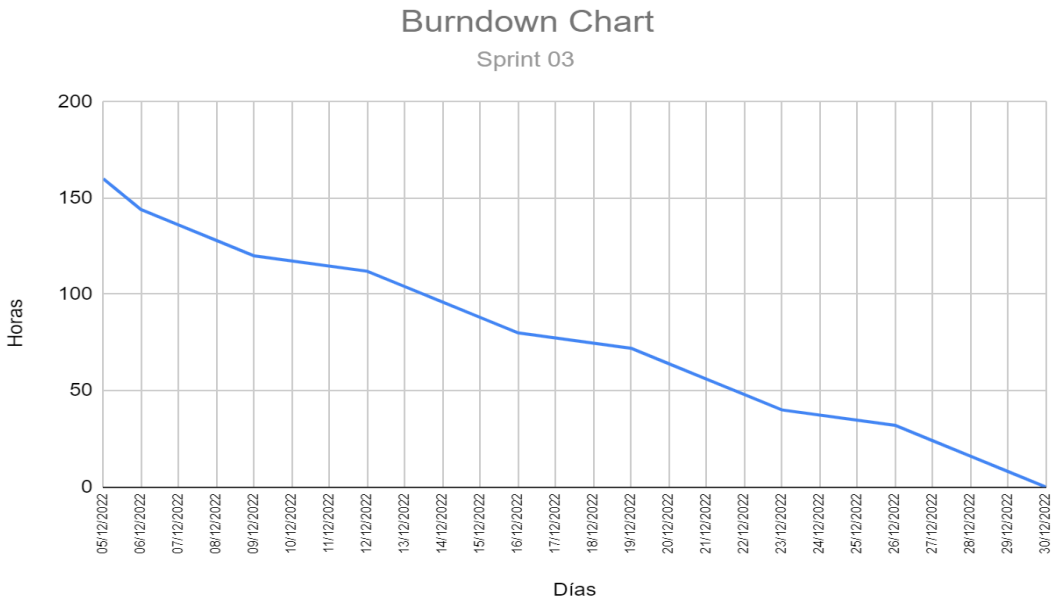
		Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
		3	05/12/2022	8 horas	30/12/2022	05/12/2022	06/12/2022	07/12/2022	08/12/2022	09/12/2022	12/12/2022	13/12/2022	14/12/2022	15/12/2022	16/12/2022	19/12/2022	20/12/2022	21/12/2022	22/12/2022	23/12/2022	26/12/2022	27/12/2022	28/12/2022	29/12/2022	30/12/2022
			Tareas Pendientes			12	9	9	9	9	9	9	9	9	8	8	7	7	5	5	3	3	1	1	0
			Horas Pendientes			160	144	136	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
H.U. 03	Preparación del entorno en el servidor	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	16														8	8					
H.U. 03	Subida de archivos	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	4																4				
H.U. 03	Despliegue de aplicación	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	8																4	4			
H.U. 03	Pruebas de API en el servidor	Pruebas	Mishell Castillo y Kevin Chuquitarco	Finalizado	8																	4	4		

		Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V	L	M	X	J	V
		3	05/12/2022	8 horas	30/12/2022	05/12/2022	06/12/2022	07/12/2022	08/12/2022	09/12/2022	12/12/2022	13/12/2022	14/12/2022	15/12/2022	16/12/2022	19/12/2022	20/12/2022	21/12/2022	22/12/2022	23/12/2022	26/12/2022	27/12/2022	28/12/2022	29/12/2022	30/12/2022
			Tareas Pendientes			12	9	9	9	9	9	9	9	9	8	8	7	7	5	5	3	3	1	1	0
			Horas Pendientes			160	144	136	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0
H.U. 03	Configuración de CORS	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	12																		4	8	
	Esfuerzo Restante				160	144	136	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0	0
	Tendencia ideal				160	152	144	136	128	120	112	104	96	88	80	72	64	56	48	40	32	24	16	8	0

Burndown Chart. En la Figura 13: Burndown Chart - Sprint 03, se muestra el avance realizado en el tiempo estimado para el desarrollo del presente sprint, donde en el eje X se muestran las fechas de los días especificados en la Tabla 17, en este caso el intervalo de tiempo inicia el 05/12/2022 y termina el 30/12/2022, en el eje Y en cambio se muestra el número total de horas estimadas al inicio, el cual se obtiene multiplicando el total de días estimado por las horas que se trabajará por día, que para este Sprint son 20 días y 8 horas diarias, lo que nos da un valor de 160 horas y que será el valor máximo de este eje, y que conforme avancen los días, el valor de horas debe ir disminuyendo con el objetivo de llegar a cero, completando el Sprint.

Figura 13

Burndown Chart - Sprint 03



Resultados del Sprint. En esta sección se explican brevemente el proceso realizado y los resultados más relevantes que se obtuvieron durante la ejecución del sprint y una vez finalizado el Sprint. Lo primero que se realizó fue entrenar el modelo y/o algoritmo de Machine Learning seleccionado al finalizar el Sprint 01 (Random Forest), del cual se obtuvieron los valores 0,9139, 0,9388 y 0,9521 para las métricas de evaluación de entrenamiento accuracy, precision y recall respectivamente. Después se procedió a guardar el modelo ya entrenado, tal como se puede evidenciar en la Figura 14: Modelo entrenado y guardado.

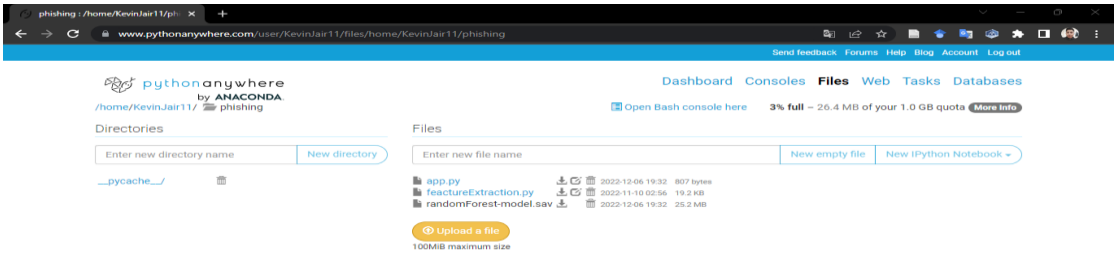
Figura 14

Modelo entrenado y guardado

En la arquitectura de la aplicación se planificó usar Python como tecnología cliente, por ello el servicio de alojamiento web seleccionado para subir la API desarrollada fue PythonAnywhere, que está basado en el lenguaje de programación Python y el framework Flask. En la Figura 15: API subido al servidor, se puede evidenciar los archivos primordiales para el funcionamiento de la API, que son: “app.py” que contiene las rutas y el código necesario para hacer la predicción, “feacturaExtraction.py” que contiene el código utilizado para la extracción de características de un sitio web y por último, el archivo “randomForest-model.sav” en donde, se guarda el modelo de Machine Learning seleccionado en el Sprint 01 mismo que se entrenó en este Sprint.

Figura 15

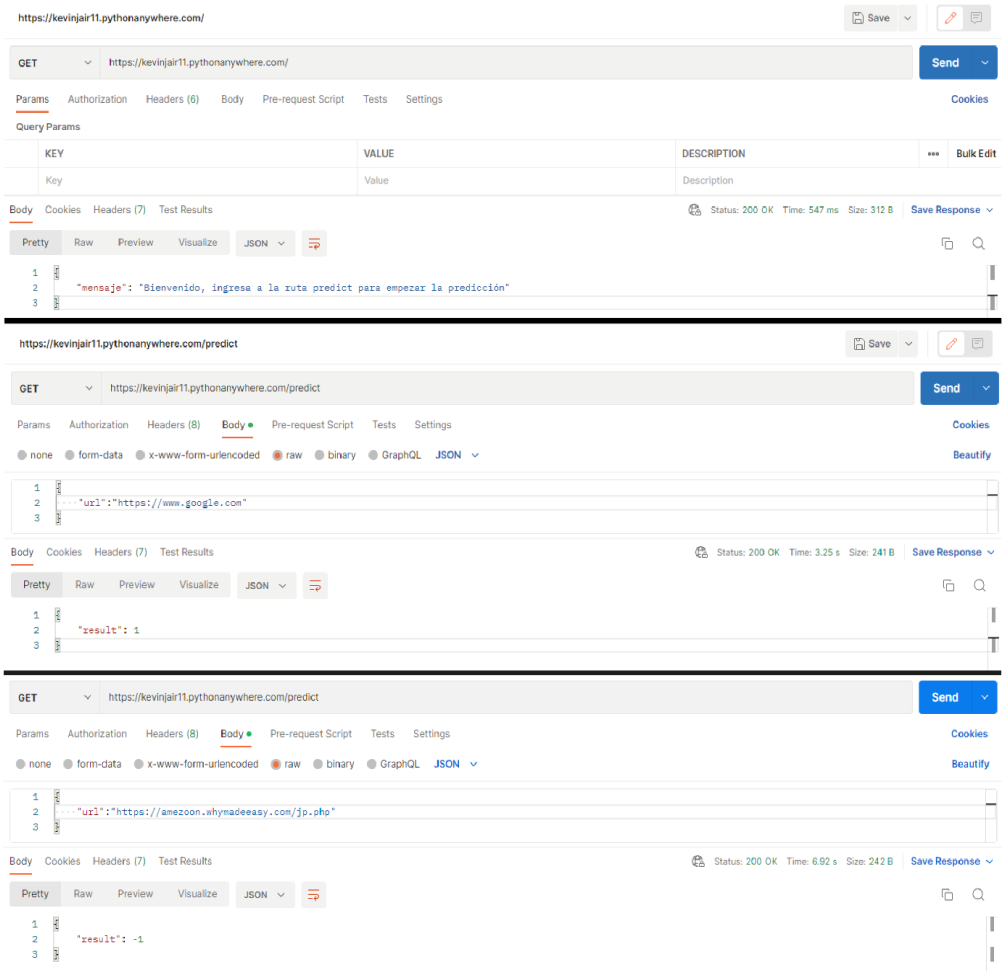
API subido al servidor



En la Figura 16: Predicción de sitios web utilizando la API desarrollada, se muestra las pruebas validadas de la API, almacenada en el servidor, que se realizó con el fin de determinar su funcionalidad, para lo cual se envió 2 URLs de prueba. La primera es del sitio web de Google retornando el valor 1 que representa a un sitio web legítimo (https://www.google.com) y la segunda de un sitio web con phishing (https://amezoon.whymadeeasy.com/jp.php) el cuál retorna un valor de -1.

Figura 16

Predicción de sitios web utilizando la API desarrollada



Nota. Al finalizar este Sprint no se encontraron inconvenientes para cumplir con la Historia de Usuario establecida, debido a ello se continuó utilizando la misma forma de trabajo.

Desarrollo de la extensión para Google Chrome

En esta sección se especifica el Sprint correspondiente para el desarrollo de la extensión de Google Chrome, que es el principal objetivo del presente proyecto. Como se definió en la arquitectura del sistema, las tecnologías usadas para el desarrollo de la aplicación Hunter Phisher fueron: HTML, CSS y JavaScript.

Sprint 04: Desarrollo de la extensión de Google Chrome

Para el desarrollo del presente Sprint, se tomó como base la Historia de Usuario H.U. 04 especificada en la Tabla 4, en primer lugar, se desarrolla la estructura básica de la extensión para Google Chrome utilizando HTML, en segundo lugar, se procede a aplicar el diseño a la estructura básica realizada para la extensión Google Chrome utilizando CSS3 y finalmente, se desarrolla un script con JavaScript que será el encargado de consumir el servicio web para la detección de sitios web con phishing utilizando modelos y/o algoritmos de Machine Learning desplegados anteriormente.

Historias de usuario detalladas. La Tabla 18: Historia de usuario para el desarrollo de la extensión de Google Chrome, presenta la Historia de Usuario H.U. 04 del sistema de detección de phishing (Hunter Phisher), donde se especifica los responsables del desarrollo, y los criterios de aceptación para la creación de la extensión de Google Chrome.

Tabla 18

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

Historias de Usuario	
Número: H.U. 04	**Usuario:** Usuario de internet
Nombre historia: Desarrollo de la extensión Google Chrome	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 8	**Interacción asignada:** 1
Programadores responsables: Mishell Castillo, Kevin Chuquitarco	
Descripción: ● Como usuario quiero una extensión para el navegador Google Chrome que me informe si un sitio web contiene phishing o no.	
Validación (Criterios de aceptación):	

Historias de Usuario	
Número: H.U. 04	**Usuario:** Usuario de internet
Nombre historia: Desarrollo de la extensión Google Chrome	
Prioridad de negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados (días): 8	**Interacción asignada:** 1
Programadores responsables: Mishell Castillo, Kevin Chuquitarco	

- Se desarrollará la extensión para que sea compatible con el navegador Google Chrome.
- La extensión consumirá un servicio web para la predicción de sitios web con phishing.
- La extensión deberá capturar la URL del sitio web donde se encuentra el usuario y realizar la predicción.
- La extensión deberá mostrar un mensaje si el sitio web tiene phishing o es legítimo.

Sprint Backlog. En la Tabla 19: Sprint Backlog 04, se especifica las tareas que se realizaron para llevar a cabo el desarrollo del sprint, los responsables de realizar cada uno de estas, las respectivas fechas en las que se planificó ejecutar el sprint, el tiempo que se estimó en horas, el esfuerzo en horas que se trabajó realmente cada día y el estado en el que actualmente se encuentra cada tarea. Cabe mencionar que se presenta el sprint backlog ya finalizado.

Tabla 19

Sprint Backlog 04

	Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X
	4	02/01/2023	8 horas	11/01/2023	02/01/2023	03/01/2023	04/01/2023	05/01/2023	06/01/2023	09/01/2023	10/01/2023	11/01/2023
	Tareas Pendientes				6	4	2	2	2	1	0	0
	Horas Pendientes				64	48	32	24	16	8	0	0

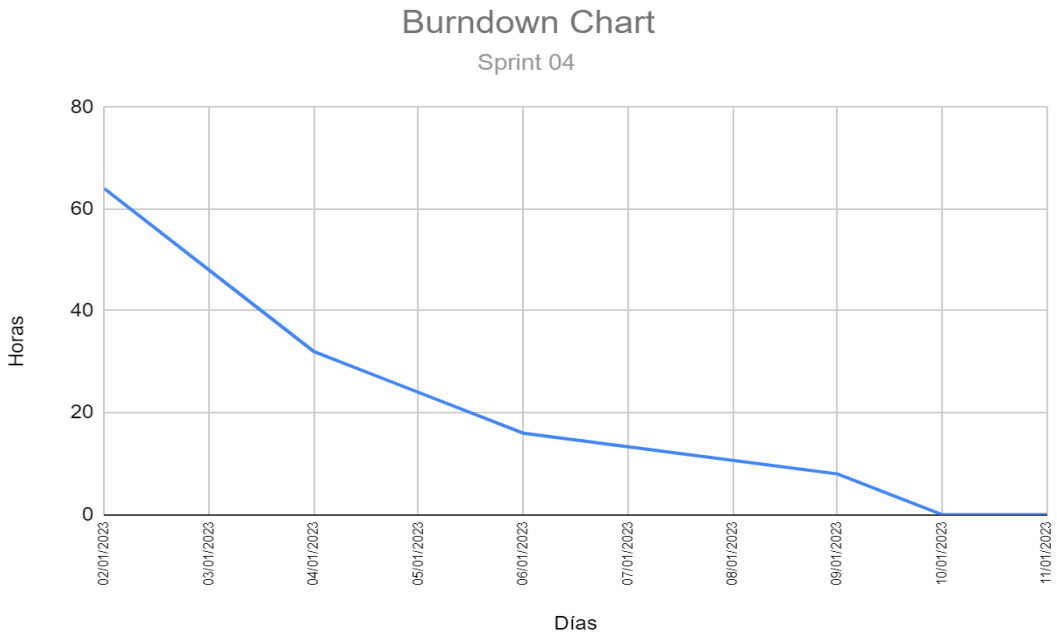
Backlog	Tarea	Categoría	Responsables	Estado	Estimación (Horas)	Esfuerzo							
H.U. 04	Creación del archivo de configuración de la extensión	Codificación	Mishell Castillo	Finalizado	8	8							
H.U. 04	Codificación de HTML	Codificación	Kevin Chuquitarco	Finalizado	8	8							
H.U. 04	Codificación de estilos CSS	Codificación	Mishell Castillo	Finalizado	8		8						

		Sprint	Inicio	Jornada	Fin	L	M	X	J	V	L	M	X
		4	02/01/2023	8 horas	11/01/2023	02/01/2023	03/01/2023	04/01/2023	05/01/2023	06/01/2023	09/01/2023	10/01/2023	11/01/2023
			Tareas Pendientes			6	4	2	2	2	1	0	0
			Horas Pendientes			64	48	32	24	16	8	0	0
H.U. 04	Codificación para la obtención del sitio web actual	Codificación	Kevin Chuquitarco	Finalizado	8		8						
H.U. 04	Codificación del script para el consumo de la API	Codificación	Mishell Castillo y Kevin Chuquitarco	Finalizado	24			8	8	8			
H.U. 04	Pruebas unitarias de la extensión	Pruebas	Mishell Castillo y Kevin Chuquitarco	Finalizado	8						8		
		Esfuerzo Restante			64	48	32	24	16	8	0	0	0
		Tendencia ideal			64	56	48	40	32	24	16	8	0

Burndown Chart. En la Figura 17: Burndown Chart - Sprint 04, se muestra el avance realizado en el tiempo estimado para el desarrollo del presente sprint, donde en el eje X se muestran las fechas de los días especificados en la Tabla 19, en este caso el intervalo de tiempo inicia el 02/01/2023 y termina el 11/01/2023, en el eje Y en cambio se muestra el número total de horas estimadas al inicio, el cual se obtiene multiplicando el total de días estimado por las horas que se trabajará por día, que para este Sprint son 8 días y 8 horas diarias, lo que nos da un valor de 64 horas y que serpa el valor máximo de este eje, y que conforme avancen los días, el valor de horas debe ir disminuyendo con el objetivo de llegar a cero, completando el Sprint.

Figura 17

Burndown Chart - Sprint 04



Resultados del Sprint. En esta sección se explican brevemente el proceso realizado y los resultados más relevantes que se obtuvieron durante la ejecución del sprint y una vez finalizado el Sprint. Lo primero que se realizó para la construcción de la extensión de Google Chrome fue la creación del archivo “manifest.json” el mismo que contiene la configuración necesaria es el nombre, la descripción, el ícono, la versión y los permisos de la extensión.

Luego se crearon los directorios para almacenar los estilos, imágenes y scripts que usará la extensión, después se realizó la estructura y el diseño de la aplicación basados en los Mockups de las Figura 4, Figura 5 y Figura 6, definidos en la etapa de diseño, y finalmente se aplicó el script correspondiente para realizar la predicción de sitios web con phishing.

En la Figura 18: Extensión de Google Chrome desarrollada, se puede evidenciar el resultado final de este Sprint, donde se muestra la ejecución de la extensión de Google Chrome en un escenario donde un sitio web es legítimo (Figura 18a), en otro donde la extensión se encuentra consumiendo el servicio web desarrollado (Figura 18b) y un último, donde el sitio web tiene phishing (Figura 18c).

Figura 18

Extensión de Google Chrome desarrollada

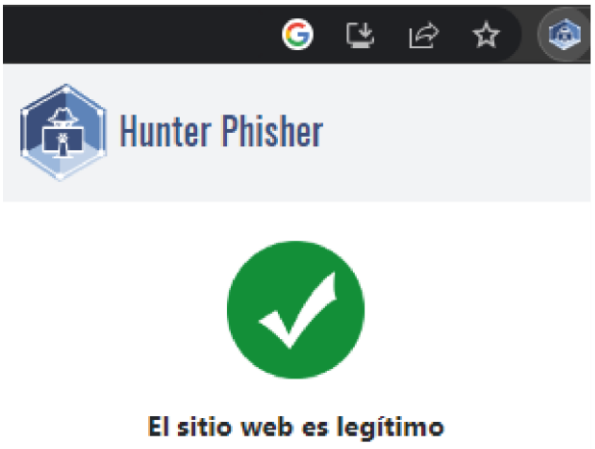


Fig. 18a

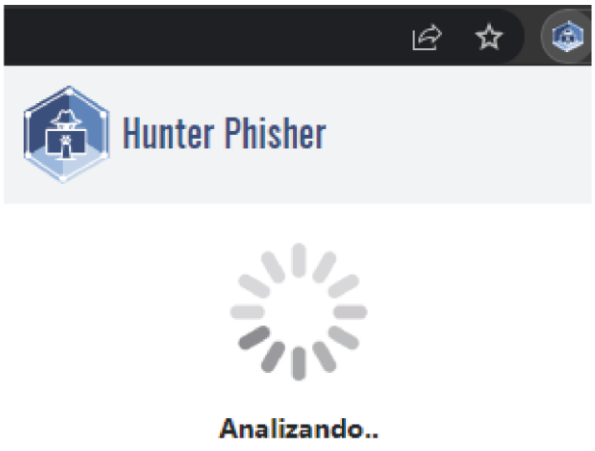


Fig. 18b

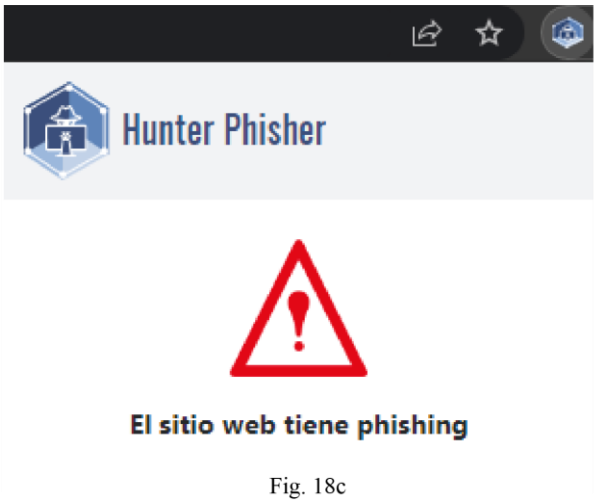


Fig. 18c

Nota. Al finalizar el último Sprint no se encontraron inconvenientes para cumplir con la Historia de Usuario establecida.

Resumen del desarrollo del sistema de detección de sitios web con phishing

Sprint 01: Se seleccionó Random Forest como modelo de Machine Learning para realizar las predicciones de sitios web con phishing o legítimos después de haber probado los 6 modelos y/o algoritmos de Machine Learning que se presentan en la Tabla 2., con el fin de garantizar una precisión alta y aceptable para predecir sitios web con phishing.

Sprint 02: Se seleccionaron 30 características que podrían ser extraídas a partir de una URL y que tenían una frecuencia igual o mayor a la media del total de frecuencias de todas las características. Además de ser probadas en grupos de 10 y combinados para encontrar los mejores valores del accuracy y garantizar una mejor precisión al momento de predecir sitios web con phishing. Además, se creó un dataset con las características seleccionadas a partir de otro que contenía solamente URLs de sitios web con phishing y legítimos.

Sprint 03: Se entrenó el modelo de Machine Learning seleccionado con el dataset creado en el anterior Sprint, se generó una API Rest y se subió a un servidor en la nube.

Sprint 04: Se desarrolló la extensión de Google Chrome utilizando las tecnologías HTML, CSS Y JavaScript.

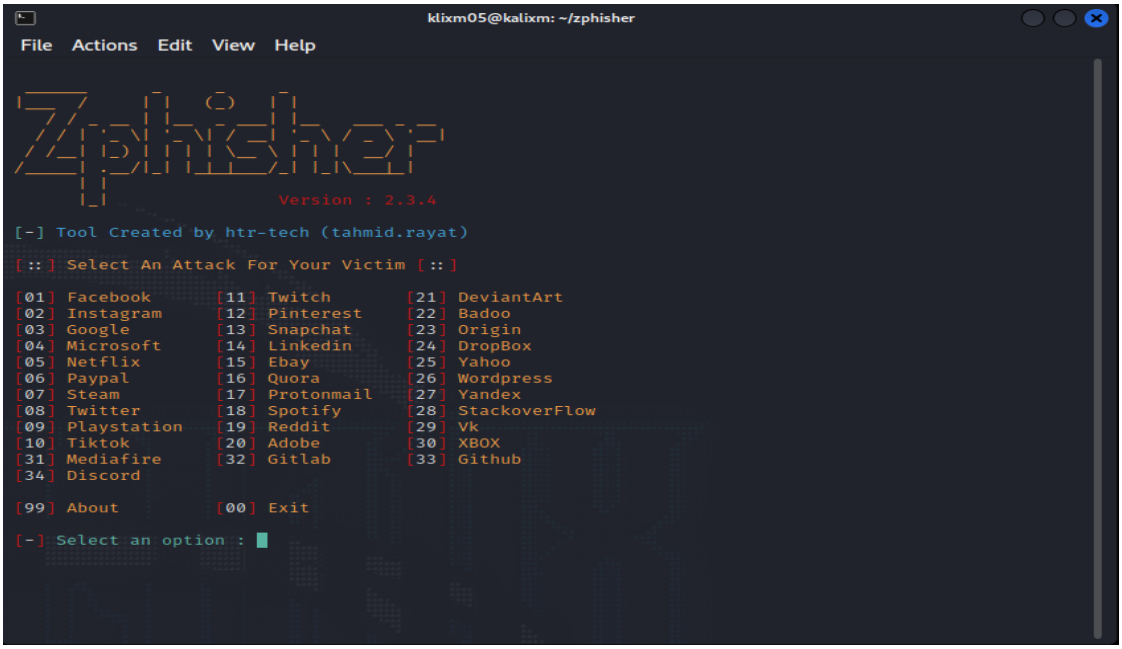
Capítulo IV

Validación Del Sistema

En este capítulo se realizan las pruebas necesarias para cumplir con la validación de la extensión de Google Chrome (Hunter Phisher). Para ello, se empleó la herramienta Zphisher, que está disponible para el sistema operativo Kali Linux, la cual sirve para generar sitios web con Phishing a través de un terminal, además, esta herramienta recopila los datos ingresados por el usuario, y finalmente muestra los datos ingresados y su dirección IP (Alshabib et al., 2022; Vira Yudha & Wisnu Wardhani, 2021)**.** En la Figura 19: Ataques disponibles Zphisher, se puede evidenciar los sitios web con Phishing que la aplicación puede generar.

Figura 19

Ataques disponibles Zphisher

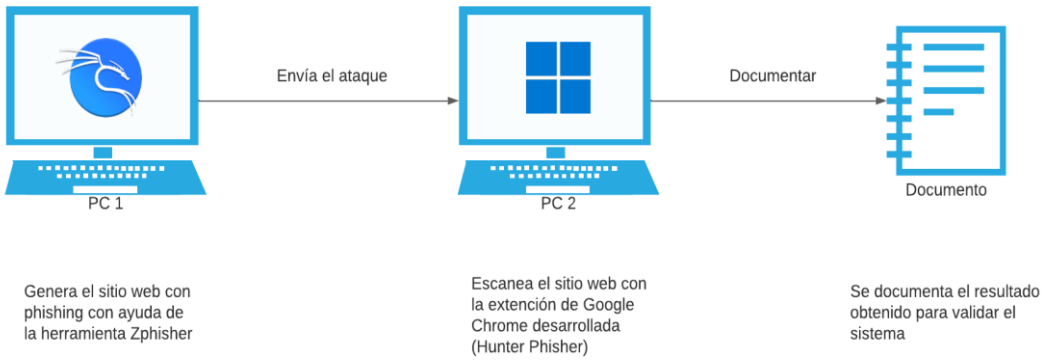


Para la realización de las pruebas se requirió de dos computadores: el primer computador tiene instalado una máquina virtual con el sistema operativo Kali Linux y es el encargado de remitir ataques Phishing, en cambio, en el segundo computador está instalado la extensión de Google Chrome (Hunter Phisher). Una vez que el ataque es enviado desde el

primer computador, se accede desde el segundo computador para inspeccionar el sitio web y documentar el resultado. En la Figura 20: Proceso de ejecución de pruebas, se muestra el proceso explicado anteriormente.

Figura 20

Proceso de ejecución de pruebas



Definición y aplicación de métricas de evaluación

Aplicación de las métricas de evaluación

Para aplicar las métricas de evaluación lo primero que se realizó fue las pruebas de la extensión Hunter Phisher con el primer modelo de Machine Learning entrenado, utilizando la herramienta ZPhisher, de los cuales se obtuvieron los resultados que se muestran en la Tabla 20: Resultados pruebas de Hunter Phisher con primer modelo de Machine Learning. La tabla está compuesta por los atributos: nombre del sitio web en general, nombre de las secciones específicas del sitio web a probar, resultado esperado y la predicción del modelo correspondiente. En esta tabla se registraron las pruebas del sistema de detección de ataques Phishing, para lo cual se utilizaron 43 sitios web con Phishing generados por la herramienta ZPhisher y 43 sitios web que pertenecen a la misma sección del sitio web pero Legítimos. Los sitios web que se seleccionó se muestra en la Figura 19 y son en total 34, mismos que son aquellos que están disponibles en Zphisher, tal como se muestra en la columna “SITIO WEB” en la Tabla 20, para una mejor comprensión de los sitios web accedidos los hemos

categorizado de acuerdo a la plataforma o conjunto de subsistemas que administran la misma funcionalidad de la siguiente manera:

- Redes sociales: Facebook, Instagram, Twitter, TikTok, Pinterest, Snapchat, Lindekin, Quora, DevianART, Badoo, Vk.
- Plataformas SSO (Single Sign-On): Google, Adobe, Microsoft, Yahoo.
- Plataforma de streaming por suscripción: Netflix.
- Plataforma de pagos on line: Paypal.
- Plataforma de distribución digital de videojuego: Steam, PlayStation, Origin, XBOX.
- Plataforma de streaming de video: Twitch.
- Portal web para vender y/o subastar on line: eBay.
- Correo electrónico: Protonmail, Yandex.
- Plataforma de streaming de música: Spotify.
- Plataforma social: Reddit, Discord.
- Servicio de alojamiento de archivos: DropBox, MediaFire.
- Sistema de gestión de sitios web: WordPress.
- Blog: StackoverFlow.
- Gestores de versiones: Github, Gitlab.

En la mayoría de estos Sitios se tomó en cuenta la sección Login Page debido a que es la primera interfaz que el usuario visualiza e ingresa sus datos privados dentro de un sistema web en general. El objetivo de los Phishers es robar estos datos para utilizarlos con fines ilegales (Hr et al., 2020b; Mohammad et al., 2014; Sönmez et al., 2018), sin embargo, para completar los 43 sitios web mencionados, se seleccionaron más de una sección de varios Sitios Web, los cuales están enfocados principalmente en secciones que requieren el ingreso de información por parte del usuario, los cuales ZPhisher puede generar.

Para entender cómo se realizaron las pruebas del sistema en este párrafo se explica brevemente el procedimiento para el primer Sitio Web elegido que es Facebook, para ello se tomaron 4 secciones, en específico, para ser probados, las cuales podrían ser obtenidos mediante la herramienta Zphisher. En primer lugar, se generaron las secciones de Sitios Web con phishing, a través de Zphisher: Tradicional Login Page (Página de inicio de sesión de Facebook), Advanced Voting Poll Login Page (Página de acceso a la encuesta de votación avanzada), Fake Security Login Page (Página de inicio de sesión de seguridad falsa), Facebook Messenger Login Page (Página de inicio de sesión de Messenger). Después se buscaron las mismas secciones del mismo Sitio Web legítimos y de todos estos se registraron la URL, tal información se muestra en el Anexo 5: Tabla de pruebas del primer modelo de Machine Learning. Seguido se probó cada URL con la extensión Hunter Phisher desarrollada con el primer modelo de Machine Learning implementado en el capítulo 3: Implementación del Sistema. De esta forma se generó y se probó con todos los demás Sitios Web con sus respectivas secciones. Finalmente, se puede observar, en la tabla 20, que el modelo únicamente detecta Sitios Web con Phishing, en cuanto a los Legítimos este sistema los considera como Phishing.

Tabla 20

Resultados pruebas de Hunter Phisher con primer modelo 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	Traditional Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Instagram	2	Advanced Voting Poll Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
	3	Fake Security Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
	4	Facebook Messenger Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
	5	Traditional Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
	6	Auto Followers Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
	7	1000 Followers Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
	8	Blue Badge Verify Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Google	9	Gmail Old Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
	10	Gmail New Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
	11	Advanced Voting Poll	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Microsoft	12	Microsoft Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Netflix	13	Netflix Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
PayPal	14	PayPal Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Steam	15	Steam Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Twitter	16	Twitter Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
PlayStation	17	PlayStation Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Tiktok	18	Tiktok Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Twitch	19	Twitch Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Pinterest	20	Pinterest Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Snapchat	21	Snapchat Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Linkedin	22	Linkedin Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Ebay	23	Ebay Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Quora	24	Quora Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Protonmail	25	Protonmail Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Spotify	26	Spotify Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Reddit	27	Reddit Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Adobe	28	Adobe Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
DeviantArt	29	DeviantArt Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Badoo	30	Badoo Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Origin	31	Origin Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
DropBox	32	DropBox Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Yahoo	33	Yahoo Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Wordpress	34	Wordpress Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Yandex	35	Yandex Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
StackoverFlow	36	StackoverFlow Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
VK	37	Traditional Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
	38	Advanced Voting Poll Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
XBOX	39	XBOX Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Mediafire	40	Mediafire Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Gitlab	41	Gitlab Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Github	42	Github Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
Discord	43	Discord Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
SITIOS WEB BIEN CLASIFICADOS			43		0	
SITIOS WEB MAL CLASIFICADOS			0		43	

Con los resultados mostrados en la Tabla 20, se procede a realizar la matriz de confusión correspondiente, como se muestran en la Tabla 21: Matriz de confusión del modelo del primer modelo de ML.

Tabla 21

Matriz de confusión del primer modelo de ML

	POSITIVOS	NEGATIVOS
POSITIVOS	**43 (VP)**	**43 (FP)**
NEGATIVOS	**0 (FN)**	**0 (VN)**

A continuación, se aplican las fórmulas para calcular las métricas Accuracy, Precision y Recall, referenciadas en la Tabla 3, y los resultados se muestran en la Tabla 22: Métricas de evaluación calculadas, en donde se puede determinar que la tasa de detección de sitios web

con phishing es sumamente baja, es decir: Sitios Web con Phishing y Legítimos bien clasificados con respecto a todos los datos de entrenamiento (Accuracy) tiene únicamente el 50%. Los sitios web con phishing bien clasificados con respecto a todos los sitios web clasificados como phishing (Presision) tiene un 50%, y por último se obtiene el 100 % en la métrica Recall, este valor toma en cuenta los sitios web phishing clasificados como phishing y el error de los sitios web con phishing clasificados como legítimos. El motivo de esta inexactitud en las métricas de evaluación es causado debido a que el primer modelo de Machine Learning únicamente detecta sitios web con phishing así sean legítimos.

Tabla 22

Métricas de evaluación calculadas

MÉTRICA DE EVALUACIÓN	RESULTADO
Accuracy	0,50 (50%)
Precision	0,50 (50%)
Recall	1,00 (100%)

Identificación de errores

Una vez aplicadas las métricas de evaluación al primer modelo se pudieron determinar los siguientes errores:

- Las métricas de evaluación son sumamente bajos.
- Únicamente predice resultados de Phishing, incluso en sitios web legítimos.
- Error en la predicción.

Corrección de errores y ajuste de modelos

Corrección y primer ajuste del modelo

Con la identificación de los errores presentados en la sección 4.1.3, se propuso una solución que consistía en aumentar más sitios web con Phishing y/o Legítimos al dataset de entrenamiento con el fin de mejorar las precisiones. Obteniéndose una mejora en la métrica de

Accuracy de 1,32%, en Precision de 3,23% y en Recall de 1,24%, con respecto al primer modelo entrenado en el capítulo 3: Implementación del Sistema. Se usó el dataset del estudio de (Shahrivari et al., 2020) para aumentar los registros de Sitios Web con y sin Phishing, En el nuevo dataset se añadieron 6.155 sitios web legítimos, que corresponden al 66,92% de porcentaje de aumento, dando un total de 15.352 sitios web legítimos, de igual manera se aumentó 4.898 sitios web con phishing, equivalente al 192,38% de porcentaje de aumento, dando un total de 7.444 sitios web con phishing, el nuevo dataset está compuesto por 22.796 sitios web en total. Con este nuevo dataset se procedió a entrenar nuevamente el modelo de Machine Learning (Random Forest), el cual fue seleccionado e implementado previamente en el capítulo 3: Implementación del Sistema. Una vez entrenado el modelo se aplicó las métricas de evaluación de entrenamiento donde se obtuvieron los siguientes resultados: 0,9434 (94,34%) de Accuracy, 0,9520 (95,20%) de Precision y 0,9645 (96,45%) de Recall.

En comparación con los resultados de las métricas de evaluación en entrenamiento del primer modelo implementado en el Capítulo 3**,** registrados en la Tabla 23: Comparación de modelos sin ajustar y ajustado, se puede observar una mejora en las métricas de evaluación, con un incremento del 3,23% en la métrica Acurracy, del 1,41% en Precision y del 1,30% en Recall. Por consiguiente, una mejora en las predicciones del modelo de Machine Learning.

Tabla 23

Comparación de modelos sin ajustar y ajustado (métricas aplicadas en la etapa de entrenamiento del modelo)

	Accuracy	**Precision**	**Recall**
Modelo Sin Ajustar	91,39%	93,88%	95,21%
Modelo Ajustado	94,34%	95,20%	96,45%
% Incremento	3,23%	1,41%	1,30%

Aplicación de métricas de evaluación del modelo ajustado

Nuevamente se realizaron las mismas pruebas a la extensión Hunter Phisher con el modelo ajustado de Machine Learning (ver Anexo 6: Tabla de pruebas con el primer modelo ajustado de Machine Learning) especificadas en la sección 4.1.2: Aplicación de las métricas de evaluación, utilizando la herramienta ZPhisher, con los cuales se obtuvieron los resultados que se muestran en la Tabla 24: Resultados pruebas de Hunter Phisher con modelo de Machine Learning ajustado. En esta tabla se puede observar, que el modelo detecta Sitios Web con Phishing, dando errores en la predicción en los sitios web que pertenecen a Redes sociales, Plataformas SSO, Servicio de alojamiento de archivos y Plataforma de distribución digital de videojuego, por otro lado, en los Legítimos presenta errores en los sitios web: Plataforma de distribución digital de videojuego, Plataformas SSO y Plataforma social.

Finalmente, se puede observar que las predicciones son más cercanas a las propuestas en la literatura (Sönmez et al., 2018) con un 92,18% de accuracy y (Chapla et al., 2019) con un 91,46% de accuracy. Cabe notar que, se hace referencia específicamente a la métrica accuracy debido a que es la métrica más importante ya que permite determinar el porcentaje de sitios web legítimos y con phishing bien clasificados con respecto a todos los datos de evaluación, además es la métrica de evaluación más utilizada para validar el desempeño de un algoritmo de Machine Learning (Cubas et al., 2022).

Tabla 24

Resultados pruebas de Hunter Phisher con modelo de ML ajustado

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Facebook	1	Traditional Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	2	Advanced Voting Poll Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	3	Fake Security Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
	4	Facebook Messenger Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Instagram	5	Traditional Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
	6	Auto Followers Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	7	1000 Followers Login Page	**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
	8	Blue Badge Verify Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Google	9	Gmail Old Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
	10	Gmail New Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	11	Advanced Voting Poll	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Microsoft	12	Microsoft Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Netflix	13	Netflix Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
PayPal	14	PayPal Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Steam	15	Steam Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Twitter	16	Twitter Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
PlayStation	17	PlayStation Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Phishing**

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Tiktok	18	Tiktok Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Twitch	19	Twitch Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Pinterest	20	Pinterest Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Snapchat	21	Snapchat Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Linkedin	22	Linkedin Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Ebay	23	Ebay Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Quora	24	Quora Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Protonmail	25	Protonmail Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Spotify	26	Spotify Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Reddit	27	Reddit Login Page	**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
Adobe	28	Adobe Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
DeviantArt	29	DeviantArt Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Badoo	30	Badoo Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
Origin	31	Origin Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
DropBox	32	DropBox Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
Yahoo	33	Yahoo Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Wordpress	34	Wordpress Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Yandex	35	Yandex Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
StackoverFlow	36	StackoverFlow Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
VK	37	Traditional Login Page	**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
	38	Advanced Voting Poll Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
XBOX	39	XBOX Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Mediafire	40	Mediafire Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Gitlab	41	Gitlab Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Github	42	Github Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Discord	43	Discord Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**
SITIOS WEB BIEN CLASIFICADOS			36		40	
SITIOS WEB MAL CLASIFICADOS			7		3	

Con los resultados presentados en la Tabla 23, se procede a realizar la matriz de confusión correspondiente y se muestran los resultados en la Tabla 25: Matriz de confusión modelo ajustado.

Tabla 25

Matriz de confusión modelo ajustado

	POSITIVOS	NEGATIVOS
POSITIVOS	**36 (VP)**	**3 (FP)**
NEGATIVOS	**7 (FN)**	**40 (VN)**

A continuación, se aplican las fórmulas mostradas en la Tabla 3, y se obtienen los resultados que se muestran en la Tabla 26: Métricas de evaluación calculadas modelo ajustado.

Tabla 26

Métricas de evaluación calculadas modelo ajustado

MÉTRICA DE EVALUACIÓN	RESULTADO
Accuracy	0,8837 (88,37%)
Precision	0,9231 (92,31%)
Recall	0,8372 (83,72%)

Identificación de errores

Una vez aplicadas las métricas de evaluación al primer modelo ajustado se pudo determinar los siguientes errores:

- Las métricas de evaluación aumentaron sus valores, sin embargo, se aspira obtener valores superiores al 90% para la métrica Accuracy en el campo real, de acuerdo con la Literatura.
- Predice resultados de legítimo en sitios web legítimos y viceversa.
- Error en la predicción.

Corrección y segundo ajuste del modelo

Con la identificación de los errores presentados en la sección 4.2.3, se propuso una solución que consistía en buscar y seleccionar un nuevo dataset que contenía sitios web con

Phishing y/o Legítimos con el objetivo de aumentar la cantidad de sitios web con Phishing y sin Phishing. El dataset seleccionado se llama “Phishing Dataset” y se lo encuentra en Kaggle (*Phishing Dataset*, s. f.). En el nuevo dataset se añadieron 5.030 sitios web legítimos, que corresponden al 32.76% de porcentaje de aumento, dando un total de 20.382 sitios web legítimos, de igual manera se aumentó 3.921 sitios web con phishing, equivale al 52,67% de porcentaje de aumento, dando un total de 11.365 sitios web con phishing. El nuevo datasets está compuesto por 31.747 sitios web, con este nuevo dataset se procedió a entrenar nuevamente el modelo de Machine Learning (Random Forest), el cual fue seleccionado e implementado previamente en el capítulo 3: Implementación del Sistema. Una vez entrenado el modelo se aplicó las métricas de evaluación de entrenamiento, obteniéndose los siguientes resultados: 98,56 (98,56%) de Accuracy; 0,9846 (98,46%) de Precision y 0,9896 (98,96%) de Recall. Obteniéndose una mejora en la métrica de Accuracy de 7,85%, en Precision de 4,88% y en Recall 3,94%, con respecto al primer modelo entrenado en el capítulo 3: Implementación del Sistema.

En comparación con los resultados de las métricas de evaluación en entrenamiento del primer modelo implementado en el Capítulo 3**,** registrados en la Tabla 27: comparación de modelos sin ajustar y ajustado 2 veces, se puede observar una mejora en las métricas de evaluación, con un incremento del 7,85% en la métrica Acurracy, del 4,88% en Precision y del 3,94% en Recall. Por consiguiente, una mejora en las predicciones del modelo de Machine Learning.

Tabla 27

Comparación de modelos sin ajustar y ajustado (2 veces) – métricas aplicadas en la etapa de entrenamiento del modelo

	Accuracy	**Precision**	**Recall**
Modelo Sin Ajustar	91,39%	93,88%	95,21%

	Accuracy	Precision	Recall
Modelo Ajustado 2 veces	98,56%	98,46%	98,96%)
% Incremento	7,85%	4,88%	3,94%

Aplicación de métricas de evaluación del modelo ajustado (2° ajuste)

Nuevamente se realizaron las mismas pruebas a la extensión Hunter Phisher con el modelo ajustado (2° ajuste) de Machine Learning especificadas en la sección 4.1.2: Aplicación de las métricas de evaluación, utilizando la herramienta ZPhisher, con los cuales se obtuvieron los resultados que se muestran en la Tabla 28: Resultados pruebas de Hunter Phisher con modelo de ML ajustado (2° ajuste). En esta tabla se puede observar, que el modelo detecta Sitios Web con Phishing, dando errores en la predicción en los sitios web que pertenecen a Redes sociales, Plataformas SSO, Plataforma de distribución digital de videojuego y Plataforma de streaming de música, por otro lado, en los Legítimos presenta errores en los sitios web que pertenecen a Plataforma social. De los resultados obtenidos se puede evidenciar que los errores se presentan, en Redes Sociales, Plataformas SSO, Plataforma de distribución digital de videojuegos y Plataformas sociales, en ambos ajustes (primero y segundo ajuste). Sin embargo, para el segundo ajuste del modelo de ML se presenta un menor número de errores en comparación con el primer modelo de ML (sin ajustar) y el primer ajuste de modelo de ML.

Finalmente, se puede observar que las predicciones son aún más cercanas a las propuestas en la literatura (Sönmez et al., 2018) con un 92,18% de accuracy y (Chapla et al., 2019) con un 91,46% de Accuracy. Cabe notar que, se hace referencia específicamente a la métrica Accuracy porque es la métrica más importante, ya que esta permite determinar el porcentaje de sitios web legítimos y con phishing bien clasificados con respecto a todos los datos de evaluación, además es la métrica de evaluación más utilizada para validar el desempeño de un algoritmo de Machine Learning (Cubas et al., 2022).

Tabla 28

Resultados pruebas de Hunter Phisher con modelo de ML ajustado (2° ajuste)

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Facebook	1	Traditional Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	2	Advanced Voting Poll Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
	3	Fake Security Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	4	Facebook Messenger Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Instagram	5	Traditional Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	6	Auto Followers Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	7	1000 Followers Login Page	**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
	8	Blue Badge Verify Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Google	9	Gmail Old Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	10	Gmail New Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
	11	Advanced Voting Poll	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
Microsoft	12	Microsoft Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Netflix	13	Netflix Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
PayPal	14	PayPal Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Steam	15	Steam Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Twitter	16	Twitter Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
PlayStation	17	PlayStation Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Tiktok	18	Tiktok Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Twitch	19	Twitch Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Pinterest	20	Pinterest Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Snapchat	21	Snapchat Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Linkedin	22	Linkedin Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Ebay	23	Ebay Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Quora	24	Quora Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Protonmail	25	Protonmail Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Spotify	26	Spotify Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
Reddit	27	Reddit Login Page	**Phishing**	**Phishing**	**Legítimo**	**Phishing**

SITIO WEB	ORD.	SECCIÓN DEL SITIO WEB	PRUEBAS SITIOS WEB PHISHING		PRUEBAS SITIOS WEB LEGÍTIMOS	
			RESULTADO ESPERADO	PREDICCIÓN	RESULTADO ESPERADO	PREDICCIÓN
Adobe	28	Adobe Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
DeviantArt	29	DeviantArt Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Badoo	30	Badoo Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Origin	31	Origin Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
DropBox	32	DropBox Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Yahoo	33	Yahoo Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Wordpress	34	Wordpress Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Yandex	35	Yandex Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
StackoverFlow	36	StackoverFlow Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
VK	37	Traditional Login Page	**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
	38	Advanced Voting Poll Login Page	**Phishing**	**Legítimo**	**Legítimo**	**Legítimo**
XBOX	39	XBOX Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Mediafire	40	Mediafire Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Gitlab	41	Gitlab Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Github	42	Github Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
Discord	43	Discord Login Page	**Phishing**	**Phishing**	**Legítimo**	**Legítimo**
SITIOS WEB BIEN CLASIFICADOS			38		42	
SITIOS WEB MAL CLASIFICADOS			5		1	

Con los resultados presentados en la Tabla 28, se procede a realizar la matriz de confusión correspondiente y se muestran los resultados en la Tabla 29: Matriz de confusión modelo ajustado (2° ajuste).

Tabla 29

Matriz de confusión modelo ajustado (2° ajuste)

	POSITIVOS	NEGATIVOS
POSITIVOS	**38 (VP)**	**1 (FP)**
NEGATIVOS	**5 (FN)**	**42 (VN)**

A continuación, se aplican las fórmulas mostradas en la Tabla 3, y se obtienen los resultados que se muestran en la Tabla 30: Métricas de evaluación calculadas modelo ajustado (2 ajuste).

Tabla 30

Métricas de evaluación calculadas modelo ajustado (2° ajuste)

MÉTRICA DE EVALUACIÓN	RESULTADO
Accuracy	0,9302 (93,02%)
Precision	0,9744 (97,44%)
Recall	0,8837 (88,37%)

Análisis de resultados

Para validar los resultados obtenidos del modelo de Machine Learning implementado, se procedió a realizar 10 pruebas, en campo simulado/real con un total de 86 sitios web de prueba cada una, en donde: el campo simulado (ambiente controlado) contenía 43 sitios web generados con la herramienta Zphisher, que se pueden observar en la columna SECCIÓN DEL SITIO WEB de la Tabla 20 y el campo real (ambiente no controlado) contenía los mismos 43 sitios web pero legítimos (originales), esta información se muestra en el Anexo 7 y Anexo 8. La razón de realizar 10 veces de esta prueba es verificar que las predicciones se mantienen estables y/o con pequeñas variaciones, por ello se calcula la desviación estándar en cada métrica de evaluación, para poder comprobar que los resultados además de estar en rango

aceptable se mantienen estables, y con ello poder validar el modelo de Machine Learning implementado.

En primer lugar, se realizó las pruebas en un campo simulado y/o real con el primer modelo de Machine Learning, con los resultados obtenidos se calculó el promedio de cada métrica de evaluación, con su respectiva desviación standard, obteniendo los siguientes resultados: 50%±0 para Accuracy, 50%±0 para Precisión y 100%±0 para Recall. Cabe mencionar que calcular estas métricas de evaluación permiten evaluar con que precisión el modelo predice correctamente los resultados de los sitios web y la desviación estándar representa el valor máximo con que aumentará o disminuirá el valor promedio de la métrica de evaluación, valores que determina el aceptable de la predicción. Por otra parte, los resultados de las métricas de evaluación obtenidas en la etapa de entrenamiento fueron: 91,39%±0,0853 para Accuracy, 93,88%±0,0767 para Precision y 95,21%±0,0770 para Recall. Con estos datos se pudo determinar que existe un cambio considerable en los valores encontrados por las métricas de evaluación, demostrando que el modelo de Machine Learning no es confiable en la etapa de validación (en un campo simulado/real), debido a varias razones, algunos sitios web con Phishing son difíciles de reconocer ya que utilizan nuevas o renovadas técnicas de Inteligencia Artificial para ocultar los recursos de comprobación conocidos (Cui et al., 2018).

En Segundo lugar, se aplicó pruebas en campo simulado/real al primer modelo de Machine Learning ajustado, del cual se calculó el promedio de cada métrica de evaluación, obteniendo los siguientes resultados: 85,23%±1,94 de Accuracy; 88,96%±1,88 de Precision y 80,47%±3,32 de Recall. Por otro lado, se calculó el promedio de los resultados obtenidos al ejecutar 10 veces el primer modelo ajustado (etapa de entrenamiento) se encontró los siguientes resultados: 94,25% ± 0,0429 de Accuracy; 95,16%± 0,0554 de Precision y 96,38%± 0,0422 de Recall. Finalmente, se realizó las pruebas en campo simulado/real con el segundo modelo de Machine Learning Ajustado, con cual se calculó el promedio de cada métrica de evaluación, obteniendo los siguientes resultados: 91,98%±0,8140 de Accuracy; 96,42%±1,2092

de Precision y 87,21%±1,1628 de Recall. Por otro lado, el valor promedio de las métricas de evaluación aplicadas 10 veces al modelo ajustado (etapa de entrenamiento) los resultados fueron los siguientes: 98,52% ± 0,0484 de Accuracy; 98,43%± 0,0670 de Precision y 98,88%± 0,1761 de Recall. Estos resultados se pueden analizar de mejor manera en una tabla, ver Tabla 31: Comparación de modelos sin ajustar, primer y segundo ajuste.

Tabla 31

Comparación de modelos sin ajustar, primer y segundo ajuste – métricas de evaluación del modelo

	Etapa de entrenamiento			**Campo simulado/real**		
	Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Modelo	91,39% ±0,0853	93,88% ±0,0767	95,21% ±0,0770	50% ±0	50% ±0	100% ±0
Modelo primer Ajuste	94,25% ±0,0429	95,16% ±0,0554	96,38% ±0,0422	85,23% ±1,94	88,96% ±1,88	80,47% ±3,32
Modelo Segundo Ajuste	98,52% ±0,0484	98,43% ±0,0670	98,88% ±0,1761	91,98% ±0,8140	96,42% ±1,2092	87,21% ±1,1628

Con estos resultados se puede afirmar que existe una clara diferencia entre los valores obtenidos a través de pruebas en un campo simulado/real y los valores obtenidos en entrenamiento, confirmando nuevamente lo concluido en el párrafo anterior, además de la evidente necesidad de ajustar el modelo. Como se puede observar en la Tabla 20, todos los sitios web con Phishing de prueba fueron etiquetados por el modelo como Phishing. Sin embargo, clasifica los sitios web Legítimos como Phishing, es decir, fallaba al identificar los sitios web que no contenían Phishing, por esta razón se ajustó el modelo.

Los valores obtenidos de las métricas de evaluación del segundo modelo (ajustado) con Zphisher se puede visualizar una mejora considerable, especialmente en el Acurracy (91,98%±0,8140) que se encarga de determinar el porcentaje de los sitios web con Phishing y Legítimos que son clasificados correctamente con respecto a todo el conjunto de datos de prueba. También se vio una mejora en la métrica Precisión (96,42%±1,2092) que es la encargada de determinar el porcentaje de sitios web con Phishing clasificados correctamente, con respecto a todos los sitios web que el modelo clasificó como Phishing. Con la métrica Recall (87,21%±1,1628) se observa que disminuye, esto es debido a que esta métrica se encarga de determinar el porcentaje de sitios web con Phishing clasificados correctamente con respecto a todas las instancias de sitios web únicamente etiquetados como Phishing del conjunto de datos de prueba, es decir, no toma en cuenta a los datos etiquetados como Legítimos. A diferencia de otros estudios, este estudio se probó en un campo simulado/real (a través de ZPhisher) y se obtuvo en Accuracy el valor más alto de 93,02% y el más bajo con 90,70%, valores que están aproximadamente dentro de los valores encontrados en la literatura 92,18% de accuracy en (Sönmez et al., 2018) y un 91,46% de Accuracy en (Chapla et al., 2019). De los resultados anteriores, se puede indicar que el sistema de detección de intrusos (IDS) para ataques phishing, implementado mediante modelos y/o algoritmos de Machine Learning, a través de una extensión Google Chrome, presenta resultados que están dentro del rango aceptable de predicciones según la revisión de la literatura realizada.

Capítulo V

Conclusiones y Recomendaciones

Conclusiones

A continuación, se plantean las conclusiones a las que se ha llegado en el desarrollo de este trabajo de investigación:

- El IDS desarrollado para la detección de ataques Phishing denominado “Hunter Phishing” se entrenó con un dataset de 22.796 sitios web (7.444 sitios web con Phishing y 15.352 sitios web legítimos). Para ello, se determinó y extrajo 30 características (recursos de comprobación). Se utilizaron 6 modelos y/o algoritmos de Machine Learning seleccionados de acuerdo con la revisión sistemática realizada. Se probaron y analizaron los resultados obtenidos, en cada modelo, mediante la aplicación de métricas de evaluación y de acuerdo a los resultados se eligió el algoritmo Random Forest por presentar las métricas de Accuracy y Precisión las más altas para detectar Sitios Web con Phishing, lo que permitió cumplir con el primer objetivo específico: Conocer el estado del arte sobre métodos y técnicas.
- Al finalizar el proceso de desarrollo de software, se diseñó e implementó un sistema de detección de Phishing, a través del desarrollo de una extensión para Google Chrome, empleando técnicas de Machine Learning, el cual es capaz de predecir si un sitio web contiene phishing o es legítimo, lo que permitió cumplir con el segundo objetivo específico: Implementar un sistema de detección de intrusos en sitios web, a través del desarrollo de una extensión para Google Chrome, empleando técnicas de Machine Learning.
- Para validar el IDS (Hunter Phisher) implementado, se utilizó la herramienta Zphisher para generar sitios web con Phishing de prueba, donde con el primer modelo entrenado e implementado se obtuvieron valores de las métricas de evaluación considerablemente bajas, razón por la cual, se ajustó el modelo de Machine Learning dos veces,

obteniendo los mejores resultados en las métricas de evaluación aplicadas para el segundo ajuste, lo que permitió verificar el objetivo específico 3: Validar los resultados, analizar los errores y ajustar los modelos del sistema de detección de intrusos.

- La aplicación de la metodología Scrum, resulto de gran ayuda para cumplir con los objetivos de este proyecto, debido a los eventos que contiene esta metodología apoyaron el desarrollo flexible, para planificar las tareas, solventar errores y mejoras a futuro en cada Sprint. El desarrollo de software implica coordinación y trabajo en equipo. Este proyecto aportó en su gran mayoría a tener experiencia en lo que respecta trabajo en equipo, además de la importancia de la comunicación que debe existir entre todos los participantes para que el desarrollo de un proyecto fluya sin mayores complicaciones, también es recomendable siempre usar una metodología de desarrollo de software con el fin de organizar las tareas y así cumplir los objetivos planteados al iniciar el proyecto.
- Finalmente, se puede concluir que la extensión de Google Chrome desarrollada (Hunter Phisher) puede ser puesta en un entorno real siempre y cuando exista un mecanismo el cual se encargue de recolectar en forma periódica nuevos sitios web con phishing para alimentar el dataset, con el fin de adaptarse a nuevos ataques de phishing en sitios web.

Recomendaciones

- Al momento de realizar una revisión de la literatura, se recomienda la selección de artículos relevantes, con el fin de obtener buenos fundamentos para iniciar una investigación.
- Para la selección de los recursos de comprobación para la detección de phishing en sitios web, se recomienda realizar diferentes escenarios, con la finalidad de determinar el aporte de cada uno de ellos.
- Para la creación de un dataset para la detección de phishing en sitios web, se recomienda primero comprobar que las URL's estén disponibles, debido a que la mayoría de los sitios web con phishing, son sitios web temporales que son eliminados una vez efectuada la estafa.
- Se recomienda el uso de la metodología ágil Scrum para el desarrollo de software, debido a la flexibilidad que nos brinda al momento de desarrollar software y así alcanzar los objetivos del proyecto con ayuda de todos los integrantes del equipo.
- Para la redacción de las historias de usuarios, se recomienda tener presentes los criterios de aceptación, porque ayudan a comprender lo que se debe hacer y lo que no se debe hacer y así comprobar si el requerimiento se logró desarrollar con calidad.
- Para todo proyecto de desarrollo de software, se recomienda la creación de un repositorio para el control de versiones en el proyecto, y así mejorar la colaboración de todos los integrantes que participan en el proyecto.
- Para los desarrolladores, se recomienda la realización de pruebas unitarias, con la finalidad de comprobar que lo realizado funciona bien, y así evitar futuros inconvenientes en la etapa de pruebas.

- Para el proceso de pruebas de un modelo de Machine Learning, se recomienda realizar varias pruebas, con la finalidad de obtener la varianza y así determinar cuál es el valor máximo y cuál es el valor mínimo que llegan las métricas de evaluación aplicadas.

Bibliografía

Adi, P. (2015). Scrum Method Implementation in a Software Development Project Management. *International Journal of Advanced Computer Science and Applications*, *6*(9). https://doi.org/10.14569/IJACSA.2015.060927

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.

Akinsola, J. E. T. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*, 128-138. https://doi.org/10.14445/22312803/IJCTT-V48P126

Alam, M. N., Sarma, D., Lima, F. F., Saha, I., Ulfath, R.-E., & Hossain, S. (2020). *Phishing attacks detection using machine learning approach*. 1173-1179. Scopus. https://doi.org/10.1109/ICSSIT48917.2020.9214225

Alazab, A., Hobbs, M., Abawajy, J., & Alazab, M. (2012). *Using feature selection for intrusion detection system*. 296-301. Scopus. https://doi.org/10.1109/ISCIT.2012.6380910

Ali, W., & Malebary, S. (2020). Particle Swarm Optimization-Based Feature Weighting for Improving Intelligent Phishing Website Detection. *IEEE Access*, *8*, 116766-116780. https://doi.org/10.1109/ACCESS.2020.3003569

Alshabib, O., Jabeur, R. A., & Alserhani, F. M. (2022). DIGITAL FORENSIC ANALYTICS IN SOCIAL MEDIA ENVIRONMENT USING DNN APPROACH. . . *Vol.*, *19*.

Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine Learning from Theory to Algorithms: An Overview. *Journal of Physics: Conference Series*, *1142*, 012012. https://doi.org/10.1088/1742-6596/1142/1/012012

Anupam, S., & 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

Apps, S. C. (2022, enero 18). *Cyberattacks 2021: Statistics From the Last Year*. Spanning. https://spanning.com/blog/cyberattacks-2021-phishing-ransomware-data-breach-statistics/

Ariyadasa, S., Fernando, S., & Fernando, S. (2021). *Phishing Websites Dataset*. *1*. https://doi.org/10.17632/n96ncsr5g4.1

Arroba Medina, L. E. (2011). *Propuesta de aplicación de Scrum para minimizar los riesgos en un proyecto de desarrollo de software* [B.S. thesis]. QUITO/EPN/2011.

Bui, H.-K., Lin, Y.-D., Hwang, R.-H., Lin, P.-C., Nguyen, V.-L., & Lai, Y.-C. (2021). CREME: A toolchain of automatic dataset collection for machine learning in intrusion detection. *Journal of Network and Computer Applications*, *193*. Scopus. https://doi.org/10.1016/j.jnca.2021.103212

Chapla, H., Kotak, R., & Joiser, M. (2019). *A Machine Learning Approach for URL Based Web Phishing Using Fuzzy Logic as Classifier*. 383-388. Scopus. https://doi.org/10.1109/ICCES45898.2019.9002145

Chiew, K. L., Yong, K. S. C., & Tan, C. L. (2018). A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, *106*, 1-20. https://doi.org/10.1016/j.eswa.2018.03.050

Coronado Huamán, H. H., Han, A., Sanz García, L., Coronado Huamán, H. H., Han, A., & Sanz García, L. (2020, junio). *Detección automática de sitios web fraudulentos* [Info:eu-repo/semantics/bachelorThesis]. https://eprints.ucm.es/id/eprint/68262/

Cubas, J. E. V., Uceda, O. E. C., Niño, G. L. E. M., Adrianzén, D. J. F., & Quintana, P. H. B. (2022). Sistema de detección de phishing basado en machine learning. *Biblioteca Colloquium*. http://colloquiumbiblioteca.com/index.php/web/article/view/122

Cui, Q., Jourdan, G.-V., Bochmann, G. V., Onut, I.-V., & Flood, J. (2018). Phishing Attacks Modifications and Evolutions. En J. Lopez, J. Zhou, & M. Soriano (Eds.), *Computer Security* (pp. 243-262). Springer International Publishing. https://doi.org/10.1007/978-3-319-99073-6_12

Dushimimana, A., Tao, T., Kindong, R., & Nishyirimbere, A. (2020). Bi-directional Recurrent Neural network for Intrusion Detection System (IDS) in the internet of things (IoT). *International Journal of Advanced Engineering Research and Science*, *7*(3), 524-539. https://doi.org/10.22161/ijaers.73.68

FERNANDEZ, A. (2013). *Python 3 al descubierto—2a ed.* Alfaomega Grupo Editor.

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, *55*(1), 119-139. https://doi.org/10.1006/jcss.1997.1504

Gates, A. J., & Ahn, Y.-Y. (2017). *The Impact of Random Models on Clustering Similarity* [Preprint]. Bioinformatics. https://doi.org/10.1101/196840

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*.

Goetz, J. N., Brenning, A., Petschko, H., & Leopold, P. (2015). Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling. *Computers & Geosciences*, *81*, 1-11. https://doi.org/10.1016/j.cageo.2015.04.007

Gómez, M. R. (2021). *Curso de desarrollo Web. HTML, CSS y JavaScript. Edición 2021*. Anaya Multimedia.

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc.

Hao, J., & Ho, T. K. (2019). Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. *Journal of Educational and Behavioral Statistics*, *44*(3), 348-361. https://doi.org/10.3102/1076998619832248

Hr, M. G., Mv, A., Gunesh Prasad, S., & Vinay, S. (2020a). Development of anti-phishing browser based on random forest and rule of extraction framework. *Cybersecurity*, *3*(1). Scopus. https://doi.org/10.1186/s42400-020-00059-1

Hr, M. G., Mv, A., Gunesh Prasad, S., & Vinay, S. (2020b). Development of anti-phishing browser based on random forest and rule of extraction framework. *Cybersecurity*, *3*(1). Scopus. https://doi.org/10.1186/s42400-020-00059-1

Jain, A. K., & Gupta, B. B. (2018). PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning. En M. U. Bokhari, N. Agrawal, & D. Saini (Eds.), *Cyber Security* (Vol. 729, pp. 467-474). Springer Singapore. https://doi.org/10.1007/978-981-10-8536-9_44

Karabatak, M., & Mustafa, T. (2018). *Performance comparison of classifiers on reduced phishing website dataset*. 1-5. https://doi.org/10.1109/ISDFS.2018.8355357

Ken, S., & Sutherland, J. (2020). *The Scrum Guide*. Scrum Alliance.

Khan, N. M., Madhav C, N., Negi, A., & Thaseen, I. S. (2020). Analysis on Improving the Performance of Machine Learning Models Using Feature Selection Technique. En A. Abraham, A. K. Cherukuri, P. Melin, & N. Gandhi (Eds.), *Intelligent Systems Design and Applications* (pp. 69-77). Springer International Publishing. https://doi.org/10.1007/978-3-030-16660-1_7

Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, *4*(1). Scopus. https://doi.org/10.1186/s42400-021-00077-7

Khraisat, A., Gondal, I., & Vamplew, P. (2018). An anomaly intrusion detection system using C5 decision tree classifier. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11154 LNAI*, 149-155. Scopus. https://doi.org/10.1007/978-3-030-04503-6_14

Korkmaz, M., Sahingoz, O. K., & Diri, B. (2020). *Feature Selections for the Classification of Webpages to Detect Phishing Attacks: A Survey*. HORA 2020 - 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings. Scopus. https://doi.org/10.1109/HORA49412.2020.9152934

Kurnia, R., Ferdiana, R., & 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

Lakshmi, L., Reddy, M. P., Santhaiah, C., & Reddy, U. J. (2021). Smart Phishing Detection in Web Pages using Supervised Deep Learning Classification and Optimization Technique ADAM. *Wireless Personal Communications*, *118*(4), 3549-3564. Scopus. https://doi.org/10.1007/s11277-021-08196-7

Lauzon, F. Q. (2012). An introduction to deep learning. *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, 1438-1439. https://doi.org/10.1109/ISSPA.2012.6310529

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

Mahesh, B. (2019). *Machine Learning Algorithms -A Review*. https://doi.org/10.21275/ART20203995

Mehta, P. (2016). Introduction to Google Chrome Extensions. En P. Mehta (Ed.), *Creating Google Chrome Extensions* (pp. 1-33). Apress. https://doi.org/10.1007/978-1-4842-1775-7_1

Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, *25*(2), 443-458. https://doi.org/10.1007/s00521-013-1490-z

Ndichu, S., Ozawa, S., Misu, T., & 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

Petersen, P. (2022). *Neural Network Theory*. University of Vienna.

Phishing Dataset. (s. f.). Recuperado 17 de enero de 2023, de https://www.kaggle.com/datasets/shuvojitdas/phishing-dataset

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

Rashid, J., Mahmood, T., Nisar, M. W., & Nazir, T. (2020). Phishing Detection Using Machine Learning Technique. *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, 43-46. https://doi.org/10.1109/SMART-TECH49988.2020.00026

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., & Gordillo, S. (2010). From Mockups to User Interface Models: An Extensible Model Driven Approach. En F. Daniel & F. M. Facca (Eds.), *Current Trends in Web Engineering* (pp. 13-24). Springer. https://doi.org/10.1007/978-3-642-16985-4_2

Rivero, J. M., Rossi, G., Grigera, J., Robles Luna, E., & Navarro, A. (2011). From Interface Mockups to Web Application Models. En A. Bouguettaya, M. Hauswirth, & L. Liu (Eds.), *Web Information System Engineering – WISE 2011* (pp. 257-264). Springer. https://doi.org/10.1007/978-3-642-24434-6_20

Robbins, J. N. (2012). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*. O'Reilly Media, Inc.

Sahingoz, O. K., Buber, E., Demir, O., & 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., & 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

Shahrivari, V., Darabi, M. M., & Izadi, M. (2020). *Phishing Detection Using Machine Learning Techniques* (arXiv:2009.11116). arXiv. https://doi.org/10.48550/arXiv.2009.11116

Somesha, M., Pais, A. R., Rao, R. S., & Rathour, V. S. (2020). Efficient deep learning techniques for the detection of phishing websites. *Sadhana - Academy Proceedings in Engineering Sciences*, *45*(1). Scopus. https://doi.org/10.1007/s12046-020-01392-4

Song, C., Ristenpart, T., & Shmatikov, V. (2017). Machine Learning Models that Remember Too Much. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 587-601. https://doi.org/10.1145/3133956.3134077

Sönmez, Y., Tuncer, T., Gökal, H., & Avci, E. (2018). *Phishing web sites features classification based on extreme learning machine*. *2018-January*, 1-5. Scopus. https://doi.org/10.1109/ISDFS.2018.8355342

Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). SCRUM model for agile methodology. *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 864-869. https://doi.org/10.1109/CCAA.2017.8229928

Statcounter Global Stats—Browser, OS, Search Engine including Mobile Usage Share. (s. f.). StatCounter Global Stats. Recuperado 10 de octubre de 2022, de https://gs.statcounter.com/

Vira Yudha, G., & Wisnu Wardhani, R. (2021). Design of a Snort-based IDS on the Raspberry Pi 3 Model B+ Applying TaZmen Sniffer Protocol and Log Alert Integrity Assurance with

SHA-3. *2021 9th International Conference on Information and Communication Technology (ICoICT)*, 556-561. https://doi.org/10.1109/ICoICT52021.2021.9527511

Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., & Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. *IEEE Access*, *6*, 35365-35381. https://doi.org/10.1109/ACCESS.2018.2836950

Yuan, G.-X., Ho, C.-H., & Lin, C.-J. (2012). Recent Advances of Large-Scale Linear Classification. *Proceedings of the IEEE*, *100*(9), 2584-2603. https://doi.org/10.1109/JPROC.2012.2188013

Zahid, M., Mehmmod, Z., & Inayat, I. (2017). Evolution in software architecture recovery techniques—A survey. *2017 13th International Conference on Emerging Technologies (ICET)*, 1-6. https://doi.org/10.1109/ICET.2017.8281704

Anexos