



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E**  
**INDUSTRIAL**  
**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

**Tema:**

---

**MODELO DE MACHINE LEARNING PARA MITIGAR LOS FRAUDES  
INFORMÁTICOS DE PHISHING BASADOS EN LA INGENIERÍA SOCIAL  
EN LA FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E  
INDUSTRIAL**

---

**Trabajo de Integración Curricular Modalidad:** Proyecto de Investigación, presentado previo a la obtención del Título de Ingeniera en Tecnologías de la Información.

**ÁREA:** Hardware y Redes

**LÍNEA DE INVESTIGACIÓN:** Sistemas Administradores de Recursos

**AUTOR:** Fabiana Patricia Jaramillo Basantes

**TUTOR:** Ing. Rubén Eduardo Nogales Portero, Mg.

Ambato – Ecuador

marzo – 2023

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Integración Curricular con el tema: **MODELO DE MACHINE LEARNING PARA MITIGAR LOS FRAUDES INFORMÁTICOS DE PHISHING BASADOS EN LA INGENIERÍA SOCIAL EN LA FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Fabiana Patricia Jaramillo Basantes, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 de las segundas reformas al Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y el numeral 7.4 del respectivo instructivo del reglamento.

Ambato, marzo 2023

-----  
Ing. Rubén Eduardo Nogales Portero, Mg.

**TUTOR**

## AUTORÍA

El presente trabajo de Integración Curricular titulado: MODELO DE MACHINE LEARNING PARA MITIGAR LOS FRAUDES INFORMÁTICOS DE PHISHING BASADOS EN LA INGENIERÍA SOCIAL EN LA FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL, es absolutamente original, autentico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2023



---

Fabiana Patricia Jaramillo Basantes

C.C 1750711416

AUTOR

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Integración Curricular como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Integración Curricular en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2023



---

Fabiana Patricia Jaramillo Basantes

C.C 1750711416

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Integración Curricular presentado por la señorita Fabiana Patricia Jaramillo Basantes, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **MODELO DE MACHINE LEARNING PARA MITIGAR LOS FRAUDES INFORMÁTICOS DE PHISHING BASADOS EN LA INGENIERÍA SOCIAL EN LA FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 de las segundas reformas al Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y al numeral 7.6 del respectivo instructivo del reglamento. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, marzo 2023

-----  
Ing. Elsa Pilar Urrutia Urrutia, Mg.  
PRESIDENTE DEL TRIBUNAL

-----  
Ing. Dennis Vinicio Chicaiza Castillo.  
PROFESOR CALIFICADOR

-----  
Ing. Marco Vinicio Guachimboza Villalba.  
PROFESOR CALIFICADOR

## **DEDICATORIA**

*El presente proyecto esta dedicado a mis padres Patricia y Fabián quienes con su eterna paciencia, amor y esfuerzo me permitieron lograr una de mis grandes metas, gracias por enseñarme el ejemplo de perseverancia y valentía, de no tenerle miedo a las dificultades porque sé que Dios siempre esta conmigo.*

*Mis hermanos Luis Miguel y Milena por su apoyo y cariño incondicional, durante todo este camino, por estar a mi lado en todo momento.*

*Finalmente, quiero dedicar este proyecto a la memoria de mi abuelita Gloria Torres, por ser un ejemplo de lucha y dedicación, por convertirse de alguna u otra forma en un ser de luz en mi vida, que me ayuda a alcanzar mis metas y sueños.*

***Fabiana Patricia Jaramillo Basantes***

## **AGRADECIMEINTO**

*Quiero expresar un sincero agradecimiento, en primer lugar a Dios por guiarme en mi camino y por permitirme concluir con mi objetivo.*

*A mis padres Patricia y Fabián quienes son mi motor y mi mayor inspiración, que a través de su amor, paciencia, buenos valores, ayudan a trazar mi camino.*

*A mi pareja Kenlly por brindarme su apoyo incondicional, su amor y respaldo para alcanzar mis objetivos.*

*A mi cuñado Leonardo por su orientación y consejos para culminar con éxito el presente proyecto.*

*A mi tutor, Ing. Rubén Nogales, por guiarme en el desarrollo de este proyecto, compartir sus conocimientos y ser un ejemplo como profesional.*

## ÍNDICE DE CONTENIDOS

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR.....	iv
APROBACIÓN TRIBUNAL DE GRADO .....	v
DEDICATORIA .....	vi
AGRADECIMEINTO.....	vii
ÍNDICE DE CONTENIDOS .....	viii
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS .....	xii
RESUMEN EJECUTIVO .....	xiii
ABSTRACT.....	xiv
CAPÍTULO I.- MARCO TEÓRICO .....	1
1.1. Tema de investigación.....	1
1.1.1. Planteamiento del problema.....	1
1.2. Antecedentes investigativos .....	3
1.3. Fundamentación teórica.....	13
1.3.1. Seguridad informática .....	13
1.3.2. Delito informático .....	13
1.3.3. Ingeniería Social.....	13
1.3.4. Phishing.....	14
1.3.5. Phisher.....	15
1.3.6. Mecanismos de seguridad .....	17
1.3.7. Integridad de los datos .....	17
1.3.8. Suplantación de identidad .....	18
1.3.9. Machine learning.....	18
1.4. Objetivos.....	18
1.4.1. Objetivo general .....	18
1.4.2. Objetivos específicos .....	18
CAPÍTULO II.- METODOLOGÍA .....	19
1.1. Materiales .....	19
1.2. Métodos .....	20



1.2.1.	Modalidad de la investigación .....	20
1.2.2.	Población y muestra .....	21
1.2.3.	Recolección de la información.....	23
1.2.4.	Procesamiento y análisis de datos .....	34
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN .....		35
3.1.	Análisis y discusión de los resultados. ....	35
3.1.1.	Experimentación simulación de ataque Phishing.....	35
3.1.2.	Métodos de ofuscación de dominios .....	39
3.1.3.	Resultados de la simulación de ataque Phishing.....	42
3.2.	Desarrollo de la propuesta .....	44
3.2.1.	Lenguaje de desarrollo de Machine Learning .....	44
3.2.2.	Metodología de desarrollo.....	45
3.2.3.	Aplicación de la metodología de desarrollo.....	48
CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES .....		70
4.1.	Conclusiones.....	70
4.2.	Recomendaciones .....	70
REFERENCIAS BIBLIOGRÁFICAS.....		72
ANEXOS.....		76

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Conocimiento sobre el robo de identidad a través de la internet.....	23
<b>Figura 2.</b> Perspectiva ante la posibilidad de ser víctima de un ataque de Ingeniería Social.....	24
<b>Figura 3.</b> Herramienta informática frente a la prevención del robo de identidad .....	25
<b>Figura 4.</b> Conocimiento sobre Phishing.....	26
<b>Figura 5.</b> Víctimas de Phishing.....	27
<b>Figura 6.</b> Conocimiento para identificar sitios dudosos.....	28
<b>Figura 7.</b> Desconocimiento de los delitos informáticos.....	29
<b>Figura 8.</b> Importancia de la información.....	30
<b>Figura 9.</b> Utilización de algún tipo de seguridad al navegar en la internet.....	31
<b>Figura 10.</b> Disponibilidad de usar una herramienta informática.....	32
<b>Figura 11.</b> Accionar ante la presencia de un link (enlace) en un correo electrónico	33
<b>Figura 12.</b> Ranking de las páginas web más visitadas .....	35
<b>Figura 13.</b> Ranking de las páginas web más visitadas en la FISEI.....	36
<b>Figura 14.</b> Ranking de las redes sociales más utilizadas.....	36
<b>Figura 15.</b> Esquema de simulación de ataque Phishing .....	38
<b>Figura 16.</b> Resultados de la simulación del ataque Phishing .....	43
<b>Figura 17.</b> Entorno de trabajo de Trello.....	48
<b>Figura 18.</b> Esquema del proceso de desarrollo de la propuesta .....	49
<b>Figura 19.</b> Estructura de la URL .....	50
<b>Figura 20.</b> Gráfico t-SNE.....	60
<b>Figura 21.</b> Matriz de confusión y ROC del modelo RNA de los datos de testeo y validación .....	65
<b>Figura 22.</b> Estructura de la extensión Phish Alert.....	67
<b>Figura 23.</b> Extensión subida y activa en el navegador.....	68
<b>Figura 24.</b> Página Phishing – mensaje de alerta.....	68
<b>Figura 25.</b> Página Legítima – mensaje en consola.....	69
<b>Figura 26.</b> Valor del alpha de cronbach .....	83
<b>Figura 27.</b> Página clonada de la plataforma educativa de la FISEI .....	83
<b>Figura 28.</b> Página clonada del sistema integrado UTA.....	84
<b>Figura 29.</b> Página clonada de facebook .....	84
<b>Figura 30.</b> Página clonada de Netflix.....	85

<b>Figura 31.</b> Formulario de engaño de parte de Google.....	85
<b>Figura 32.</b> Formulario de engaño de parte de youtube .....	86
<b>Figura 33.</b> Funciones para la extracción de las características .....	92
<b>Figura 34.</b> Guardado del modelo RNA con la librería joblib.....	92
<b>Figura 35.</b> Función url_has_vector que retorna el vector de características de la URL .....	93
<b>Figura 36.</b> Función predecir que retorna la etiqueta predicha.....	93
<b>Figura 37.</b> API RESTful utilizando Flask.....	94
<b>Figura 38.</b> Archivo conten-scripts.js de la extensión para Chrome .....	94
<b>Figura 39.</b> Archivo manifest.json de la extensión de Chrome .....	95
<b>Figura 40.</b> Finalización de la lista de tareas .....	95

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Cadenas de búsqueda .....	3
<b>Tabla 2.</b> Criterios de inclusión y exclusión .....	6
<b>Tabla 3.</b> Tabla resumen de los estudios mencionados en la revisión bibliográfica .	11
<b>Tabla 4.</b> Encuesta a la comunidad FISEI .....	19
<b>Tabla 5.</b> Población y Muestra de la Comunidad Estudiantil .....	22
<b>Tabla 6.</b> Población y Muestra del Régimen Laboral.....	22
<b>Tabla 7.</b> Métodos de ofuscación de dominios .....	40
<b>Tabla 8.</b> Tabla comparativa de lenguajes de desarrollo de Machine Learning .....	45
<b>Tabla 9.</b> Tabla comparativa de metodologías ágiles .....	46
<b>Tabla 10.</b> Conjunto de datos utilizados .....	50
<b>Tabla 11.</b> Características de la URL.....	51
<b>Tabla 12.</b> Tabla comparativa de los algoritmos .....	63
<b>Tabla 13.</b> Alfa de Cronbach .....	76

## RESUMEN EJECUTIVO

En la actualidad los sitios web de Phishing siguen siendo una amenaza importante en el amplio ciberespacio de internet. Cuando un usuario visita una URL Phishing, los atacantes obtienen información personal y confidencial del usuario. Los estafadores informáticos emplean diferentes técnicas de Ingeniería Social para efectuar robos de identidad o lanzar ataques selectivos. Estudiantes y catedráticos no están absueltos ante la fuerte influencia de las distintas técnicas de Ingeniería Social. Los phishers buscan la manera de perjudicar y hacer dinero mediante la manipulación y extorción a los usuarios incautos. Para resolver este problema, el presente trabajo de integración curricular propone implementar un modelo de Machine Learning para la detección del Phishing desplegada como una extensión al navegador. Para el análisis de las URLs y construcción del dataset se extrajeron 24 características de su estructura. Se realizó una comparación entre varios modelos de clasificación como: Random Forest (RF), Support Vector Machine (SVM), Red Neuronal Artificial (RNA) y K-Nearest Neighbors (KNN) para elegir el más adecuado y el que mejor se ajuste al problema. Una vez analizado los algoritmos de los distintos modelos de entrenamiento, el modelo utilizado para clasificar las URLs es una red neuronal artificial (RNA) consiguiendo una exactitud del 99.98%. La finalidad del Modelo de Machine Learning incorporada en la extensión para el navegador es ayudar a la comunidad de la FISEI. La extensión mitigará y evitará que los usuarios se conviertan en víctimas de actividades maliciosas, como el caer en URLs Phishing que aplican varios de los principios de persuasión de la Ingeniería Social.

**Palabras clave:** Phishing, Ingeniería Social, extracción de características, URL, Machine Learning, extensión al navegador.

## ABSTRACT

Nowadays, Phishing websites continue to be a significant threat in the vast cyberspace of the internet. When a user visits a Phishing URL, attackers obtain the user's personal and confidential information. Cyber criminals use various social engineering techniques to carry out identity theft or launch targeted attacks. Students and professors are not exempt from the strong influence of different social engineering techniques. Phishers seek ways to harm and make money through the manipulation and extortion of unsuspecting users. To address this problem, the present curricular integration work proposes implementing a Machine Learning model for Phishing detection deployed as a browser extension. 24 characteristics of its structure were extracted for the analysis of URLs and construction of the dataset. A comparison was made between various classification models such as Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (RNA), and K-Nearest Neighbors (KNN) to choose the most appropriate and best fit for the problem. Once the algorithms of the different training models were analyzed, the model used to classify URLs is an artificial neural network (RNA) achieving an accuracy of 99.98%. The purpose of this work is to help the FISEI community. The extension will mitigate and prevent users from becoming victims of malicious activities such as falling into Phishing URLs that apply various principles of Social Engineering.

**Keywords:** Phishing, Social Engineering, feature extraction, URL, Machine Learning, browser extension.

## **CAPÍTULO I.- MARCO TEÓRICO**

### **1.1. Tema de investigación**

MODELO DE MACHINE LEARNING PARA MITIGAR LOS FRAUDES INFORMÁTICOS DE PHISHING BASADOS EN LA INGENIERÍA SOCIAL EN LA FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL.

#### **1.1.1. Planteamiento del problema**

Phishing es un delito informático que aplica Ingeniería Social para sustraer información confidencial y obtener beneficios. La palabra Phishing se deriva de la correlación de los delincuentes de internet. El Phishing usa como engaño los correos electrónicos para la obtención de contraseñas e información personal, sobre todo comprometedor de un amplio ciberespacio de internautas.[1]

Desde hace más de 20 años ya se empleaban distintas técnicas de Ingeniería Social. Los criminales engañan a personas solicitando datos confidenciales. El objetivo de la Ingeniería Social es dañar a organizaciones, ya sean grandes o pequeñas. Tal es el caso del primer ataque dirigido a la empresa estadounidense América Online.[1]

A nivel mundial, según Nadezhda Demidova investigadora de seguridad en Kaspersky Lab. En el año 2017 ha registrado ataques de Phishing lanzados contra 131 universidades en 16 países del mundo. Con 83 universidades ubicadas en los Estados Unidos, 21 en el Reino Unido, 7 en Australia y 7 en Canadá [2]. Aunque cada universidad dispone de medidas de seguridad. Los ciberdelincuentes buscan usuarios incautos fáciles de engañar para acceder a la información.

A nivel nacional, Ecuador no es la excepción, según [3] ocupa el quinto lugar de los países en América Latina más vulnerables a delitos informáticos, que usan métodos de Ingeniería Social. Ecuador es un país tradicionalista y la mayoría de los habitantes prefieren una comunicación o comercio más clásico. El uso de los medios digitales se lo efectúa de una manera muy particular y mínima en el país. La falta de

aprovechamiento y el incorrecto uso de la internet se hace evidente. Por esta razón Ecuador es un país codiciado por varios ciberdelincuentes para cometer todo tipo de ataques informáticos, ya que la población no presta interés para adquirir una capacitación correcta y adecuada para navegar en la internet. En el periodo 2018 – 2020 con datos de [4] demuestra que el Phishing es el delito informático número uno en el Ecuador. Con un promedio anual de 43.2% es decir de cada 10 delitos informáticos 4 corresponden al Phishing.

A nivel local, la Facultad de Ingeniería en Sistemas Electrónica e Industrial (FISEI) y su comunidad se encuentra en un avance continuo con respecto a la tecnología. Por consiguiente, la facultad impulsa a varios proyectos de desarrollo empresarial. Proyectos que motivan a los delincuentes informáticos al robo de información [5]. Estudiantes y catedráticos no están absueltos ante la fuerte influencia de las distintas técnicas de Ingeniería Social. Los estafadores informáticos emplean diferentes técnicas de Ingeniería Social para efectuar robos de identidad o lanzar ataques selectivos. Los atacantes buscan la manera de hacer dinero y las entidades universitarias son esa fuente. Lo que es peor, los delincuentes informáticos usan correos electrónicos de estudiantes para solicitar préstamos fraudulentos [6]. Por tal motivo es importante realizar un análisis del fraude informático Phishing, ante los principios de persuasión de Ingeniería Social en la FISEI. El presente estudio ayudará a prevenir tales incidencias con un alto impacto de negatividad para la facultad.

Otra incidencia no menos importante del fraude informático Phishing en las entidades universitarias, implica el hurto de sus bases de datos. Los datos pueden ser información privada y contenido de distintos tipos de investigación. Las investigaciones tienden a ser exclusivas e impactantes que pueden comprometer a empresas involucradas con la entidad. Por otra parte, los datos a más de ser útiles como espionaje, también tiene un valor económico. Por consiguiente, la realización de este estudio podrá disminuir el efecto negativo, causado desde los distintos puntos de vista de estudiantes, docentes, administrativos y cada una de las partes que conforma la FISIE. Por tanto, otorgará a la facultad un nivel alto de seguridad informática.



## 1.2. Antecedentes investigativos

En este trabajo, se realizó una revisión de literatura en artículos publicados en las siguientes bases de datos científicas: Scopus, IEEE Xplore, ACM Digital Library, Willey, Science Direct y Springer. Las cadenas de búsqueda se establecieron con las siguientes palabras claves: Phishing detection, Machine Learning, Websites, URL, Homoglyph, Typographic y Extensión, en el rango de los años 2018 – 2022. Por consiguiente, la **Tabla 1.** muestra las cadenas de búsqueda utilizadas y el número de artículos encontrados.

**Tabla 1.** Cadenas de búsqueda  
**Elaborado por:** Fabiana Jaramillo

	Revista	Cadena	N° artículos
<b>Cadena 1</b>	Scopus	(TITLE-ABS-KEY (“phishing detection”) AND TITLE-ABS-KEY (“machine learning”) AND TITLE-ABS-KEY (websites) OR TITLE-ABS-KEY (url) AND TITLE-ABS-KEY (extension) OR TITLE-ABS-KEY (homoglyph)) AND (LIMIT-TO (PUBYEAR, 2022) OR LIMIT-TO (PUBYEAR, 2021) OR LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018))	7
	IEEE Xplore	("All Metadata":"phishing detection") AND ("All Metadata":"machine learning") AND ("All Metadata":websites) OR ("All Metadata":url) AND ("All Metadata":extension) OR ("All Metadata":homoglyph) Filters Applied: 2018 - 2022	105
	ACM Digital Library	[[All: "phishing detection"] AND [All: "machine learning"] AND [All: websites]] OR [[All: url] AND [All: extension]] OR [All: homoglyph] AND [E-Publication Date: (01/01/2018 TO 12/31/2022)]	6086

	Wiley	""Phishing detection" AND "machine learning" AND websites OR URL AND extension OR homoglyph" anywhere Applied Filters: 2018 - 2022	6042
	Science Direct	"phishing detection" AND "machine learning" AND websites OR url AND extension OR homoglyph Refine by: 2018 -2022	4209
	Springer	"phishing detection" AND "machine learning" AND websites OR url AND extension OR homoglyph' within 2018 - 2022	11199
<b>Cadena 2</b>	Scopus	( TITLE-ABS-KEY ( "phishing detection" ) AND TITLE-ABS-KEY ( "machine learning" ) AND TITLE-ABS-KEY ( websites ) OR TITLE-ABS- KEY ( url ) AND TITLE-ABS-KEY ( homoglyph ) OR TITLE-ABS-KEY ( typographic ) AND TITLE-ABS-KEY ( "hash function" ) ) AND ( LIMIT-TO ( PUBYEAR , 2022 ) OR LIMIT-TO ( PUBYEAR , 2021 ) OR LIMIT-TO ( PUBYEAR , 2020 ) OR LIMIT-TO ( PUBYEAR , 2019 ) OR LIMIT-TO ( PUBYEAR , 2018 ) )	1
	IEEE Xplore	("All Metadata": "phishing detection") AND ("All Metadata": "machine learning") AND ("All Metadata": websites) OR ("All Metadata": URL) AND ("All Metadata": homoglyph) OR ("All Metadata": typographic) AND ("All Metadata": "hash function") Filters Applied: 2018 - 2022	69
	ACM Digital Library	[[All: "phishing detection"] AND [All: "machine learning"] AND [All: websites]] OR [[All: url] AND [All: homoglyph]] OR [[All: typographic]	85

		AND [All: "hash function"]] AND [E-Publication Date: (01/01/2018 TO 12/31/2022)]	
	Wiley	"phishing detection" AND "machine learning" AND websites OR URL AND homoglyph OR typographic AND "hash function" anywhere Applied Filters: 2018 - 2022	28
	Science Direct	"phishing detection" AND "machine learning" AND websites OR URL AND homoglyph OR typographic AND "hash function" Refine by: 2018 -2022	111
	Springer	"phishing detection" AND "machine learning" AND websites OR URL AND homoglyph OR typographic AND "hash function" within 2018 - 2022	26
<b>Total</b>			27968

El fin de la revisión de literatura es describir el modelo de Machine Learning para detectar sitios web Phishing aplicando una extensión o complemento en el navegador. En este contexto, la revisión de literatura da como resultado 27968 artículos investigativos, donde se encuentran las palabras claves establecidas. De los 27968 artículos se eliminaron 13850 artículos que solicitaron un pago extra. Como resultado se tiene 13864 artículos restantes. Los 13864 se excluyeron porque no se relacionaban con la detección del Phishing con Homoglifos o Tipograficos aplicando Machine Learning. Dejando un total de 254 artículos restantes de los cuales se discriminaron 217. Los 217 son discriminados porque las metodologías del procedimiento no se describen y no muestra resultados de validación. Del análisis anterior quedan 37 artículos. De los 37 artículos, se trabajó con 7 artículos que cumplen todos los criterios de inclusión necesarios para la investigación del estudio. **La Tabla 2.** muestra los criterios de inclusión y exclusión utilizados para seleccionar los artículos más relevantes para la investigación.

**Tabla 2.** Criterios de inclusión y exclusión

**Elaborado por:** Fabiana Jaramillo

Inclusión	<ul style="list-style-type: none"><li>- Artículos que usan modelos de Machine Learning para la detección de Phishing en sitios web.</li><li>- Artículos que emplean métodos de detección de URLs Phishing y URLs legítimas.</li><li>- Artículos relacionados a ataques de Phishing con homoglifos o Tipográficos.</li><li>- Artículos que aplican una extensión para el navegador.</li></ul>
Exclusión	<ul style="list-style-type: none"><li>- No describen sus técnicas ni resultados de validación.</li><li>- Artículos que solicitaron un pago extra.</li><li>- No emplean modelos de Machine Learning para detectar Phishing.</li><li>- Implementan detección de phishing para correos.</li></ul>

En [7] propone un nuevo enfoque que puede predecir y detectar los glifos (conocidos como caracteres homogéneos) con un alto nivel de precisión. El problema de la existencia de los homoglifos es interesante. Dado que los homoglifos tienen un aspecto similar, si no idéntico, pero con diferentes códigos de caracteres, que pueden ser utilizados para crear cadenas de texto parecidas y elaborar varios ataques basados en el engaño. Este tipo de ataques afectan a los nombres de dominios de los sitios web, a las tiendas de aplicaciones y a casi cualquier sistema informático en el que los usuarios pueden enviar contenido. El enfoque de este proyecto se inspira en el comportamiento humano de comparación visual, al intentar ver la diferencia entre dos homoglifos. En primer lugar, extraen una lista de glifos que admite un determinado tipo de letra. Para cada glifo crean cuatro mapas HitZone, con 16, 64, 100 y 256 zonas pares respectivamente. Cada zona tiene un número asignado que se registra como “hit” y si esta vacío se registra como un “fallo”. Se almacena los cuatro mapas HitZone, el código del carácter y el nombre de la fuente en una base de datos. Para calcular el porcentaje de similitud de dos glifos en un nivel de granularidad dado, recuperan los mapas HitZone asociados para ambos glifos. Por último, utilizan el cálculo del porcentaje de similitud para filtrar los glifos que no son mayor o igual que el porcentaje de similitud especificado. Como resultados de la precisión del algoritmo de predicción

de homógrafos, de la "letra mayúscula latina A", tienen una similitud del 90%, para la fuente Arial, en el nivel 4 de granularidad. Con la técnica Seekback los resultados son coherentes con su predicción, de la "letra mayúscula latina A", tienen una similitud igual o superior al 75%, para la fuente Courier New, con una granularidad de nivel 4 y 1 nivel de Seekback.

En [8] propone un marco de detección de Phishing basado en la similitud visual, con la ayuda de una red neuronal convolucional (CNN) de tripletes. La similitud visual es un factor clave en la detección de páginas de Phishing de día cero. El estudio trabaja con un conjunto de datos VisualPhish, con 155 sitios web y 9363 capturas de pantalla. El conjunto de datos facilita la detección de Phishing visual de una manera válida. A diferencia de trabajos anteriores, en lugar de hacer coincidir únicamente una página de Phishing a su homóloga legítima, generalizan la similitud visual para detectar las páginas no vistas dirigidas a los sitios web de confianza. Con esto VisualPhishNet aprende un perfil visual de los sitios web mediante una métrica de similitud entre páginas webs iguales a pesar de tener contenidos diferentes. Basándose en el análisis cualitativo de los casos exitosos, la red identifica páginas de Phishing con mucha facilidad. Páginas altamente similares a las de entrenamiento y páginas parcialmente copiadas y ofuscadas. VisualPhishNet hace frente a la gama de posibles ataques de evasión. El trabajo demuestra un salto en el rendimiento respecto a enfoques anteriores de similitud visual en 56% en la coincidencia, precisión y 30% en el ROC (Receiver Operating Characteristic) área de clasificación bajo la curva.

En [9] propone un marco llamado "ShamFinder". ShamFinder es un esquema automatizado para detectar homógrafos. El esquema puede utilizarse para tomar contramedidas directas contra el ataque e informar a los usuarios sobre el contexto de un homógrafo de IDN (nombre de dominio internacionalizado). Aunque la amenaza que suponen los ataques de homógrafos IDN no es nueva. El reciente aumento de la adopción de IDN tanto en los registros de nombres de dominio como en los navegadores web, ha provocado que la amenaza de estos ataques se extienda cada vez más. Dando lugar a ataques de phishing a gran escala, como los dirigidos a empresas de intercambio de criptomonedas. El proceso de desarrollo incluye en primer lugar, la recopilación de los nombres de dominio registrados/activos de cada TLD (top-level

domains) o listas de nombre de dominio disponibles públicamente/comerciales. A continuación, extraen los IDN de los nombres de dominio recopilados buscando los que empiezan por el prefijo “xn--”. Para encontrar homógrafos de IDN, utilizan una lista de nombres de dominio populares. Como listas de ranking de sitios web como Alexa Top Sites o Majestic Million. Por último, aprovechan las bases de datos de homoglifos combinadas SimChar y UC, para identificar posibles homógrafos de IDN. Los IDN se extraen con el mismo número de caracteres. Si dos letras correspondientes coinciden, pasan al siguiente par de letras. Si las letras no coinciden, comprueban el par figura en la base de datos de homoglifos. Si aparecen en la lista se pasa al siguiente y se repite el mismo proceso. Si encuentran letras que no coincide, se concluye que el IDN no es un homógrafo. Este trabajo aproximadamente detecta ocho veces más homógrafos que otros estudios realizados.

En [10] determina que el Phishing sigue siendo una importante amenaza para la seguridad en el ciberespacio. Los atacantes roban información mediante la presentación de un sitio falso que parece ser un clon visual de un sitio legítimo. Esta similitud visual se aplica a varios caracteres Unicode que son idénticos a los caracteres ASCII y se los conoce como homoglifos. Ante este problema, el artículo propone un modelo de detección de ataques de homoglifos que combina una función hash y aprendizaje automático. El modelo consta de dos fases, desarrollo y despliegue. La fase de desarrollo comienza primero con el preprocesamiento, un método para convertir los datos sin procesar en un conjunto de datos limpios para ser útil al análisis posterior. Como segundo paso, la extracción de características convierte los datos sin procesar en un conjunto de características con datos que constan de 70000 URL combinadas mediante la extracción de 35000 URL legítimas de Alexa y 35000 URL ilegítimas extraídas de PhishTank y PhishStat. Además, generan un conjunto de datos de homoglifos para cubrir todas las posibilidades de sitios web de Phishing y homoglifos. Como tercer paso, la selección de funciones utiliza un subconjunto del conjunto original de funciones para obtener un subconjunto más pequeño que se puede utilizar para modelar el problema. El resultado del proceso de selección de características es el conjunto de datos con un conjunto reducido de características que pueden representar mejor el problema. Luego, utilizaron varios clasificadores para entrenar el modelo. Por último, la fase de evaluación representa la efectividad del

modelo propuesto para resolver el problema de los homógrafos. Logran una precisión del 99.8% con Random Forest y la función hash, que es un proceso de conversión de texto sin formato en texto codificado de longitud fija. La función hash mejora la precisión de la detección de ataques de homógrafos.

En [11] determina que actualmente la evolución de los sitios web de Phishing han ido mejorando para poder engañar a los usuarios y evadir la detección. Por lo que, el trabajo propone un estudio de medición sobre el uso ilegal de dominios de Phishing. Para la búsqueda de páginas de Phishing, escanearon cinco tipos de dominios ofuscados en más de 224 millones de registros DNS e identificaron 657000 dominios, con la fusión de los conjuntos de datos de PhishTank y Alexa. En total suplantaron 702 marcas populares. Los cinco tipos de dominios ofuscados son: homógrafo, error de tipografía, Bits, Combo, Wrong TLD. Luego, construyen un clasificador de aprendizaje automático para detectar Phishing tanto de la web como de páginas móviles bajo squatting domains. El clasificador se basa en una medición cuidadosa de los comportamientos evasivos de las páginas de Phishing. Para superar la gran ofuscación de contenido de los atacantes, presentan nuevas características de análisis visual y OCR (Reconocimiento Óptico de Caracteres). En total, se descubrió y verificó 1175 páginas ilegales de phishing. Mostrando que estas páginas de Phishing se utilizan para varias estafas dirigidas y son altamente efectivas para evadir la detección.

En [12] señala que los caracteres visualmente similares u homógrafos, pueden ser utilizados para ataques de ingeniería social o para evadir los detectores de plagio. Por lo tanto, es importante comprender las capacidades de un ataque para identificar homógrafos. El estudio propone un modelo de deep learning que utiliza embedding learning, transfer learning, and augmentation para determinar la similitud visual de los caracteres y así identificar posibles homógrafos. El data augmentation o aumento de datos, se da por la falta de etiquetas para algunos caracteres. En su lugar, generan etiquetas débiles aprovechando el hecho de que, en general los caracteres no son homógrafos y pertenecen a sus propias clases de triviales. Esto les permite muestrear caracteres de diferentes clases mediante una elección uniforme y aleatoria, con una mayor fiabilidad. Para el entrenamiento de la embedding function (función de incrustación) hacen uso de la versión modificada de la función Triplet Loss. En la

función Triple Loss, un triplete se compone por un punto de datos ancla, un punto de datos positivo de la misma clase que el ancla, y un punto de datos negativo de una clase diferente. Para producir sus propios tripletes, utilizaron las etiquetas débiles para generar un ancla y una muestra positiva del mismo punto de código, con una muestra negativa de un punto de código diferente. Cada uno de estos conjuntos de datos se asignan a distintos clusters. Para calcular la pérdida de tripletes utilizaron la similitud del coseno. El método de transfer learning (aprendizaje por transferencia), para embedding function, utilizan EfficientNet sin su capa de clasificación softmax. EfficientNet es una red neuronal convolucional (CNN) y esta preentrenado en ImageNet. Este modelo supera con creces la distancia de compresión normalizada en la identificación de homógrafos por pares, logrando una precisión media de 97%.

En [13] señala que, el Phishing es uno de los ataques más extendidos basados en la Ingeniería Social. El estudio demuestra que un modelo de aprendizaje automático entrenado con conjuntos de datos recogidos hace algunos años, podría tener un alto rendimiento, pero su rendimiento disminuye notablemente con conjuntos de datos actuales, utilizando en ambos casos las mismas características. Para confirmar estas afirmaciones, crean un nuevo conjunto de datos, Phishing Index (PILU-60K), que contiene 60.000 URL de índice y de inicio de sesión legítimas. Evaluaron varios métodos de aprendizaje como: Support Vector Machines (SVM), Random Forest (RF), k-Nearest Neighbours (kNN), Naive Bayes (NB) and Logistic Regression (LR). Para la extracción de características en los conjuntos de datos en cada URL se extrae 38 descriptores que incluyen reglas de URL y características de Natural Language Processing (NLP) de símbolos en la URL, dígitos en el dominio, el subdominio y en la ruta, longitudes de sus distintas partes, nivel de subdominio, aleatoriedad del dominio, TLD conocido, www o com, métricas de palabras como máximo, mínimo, media, desviación estándar, número de palabras, palabras compuestas, palabras iguales o similares a una marca famosa o a una palabra clave como "secure" o "login", caracteres consecutivos en la URL y punycode. El archivo de características se introduce en un script de Python3 que utiliza scikit learn para dividir el conjunto de datos, entrenar, probar los modelos y obtener los resultados medidos con precisión y F1-Score. Por último, descubrieron que el algoritmo SVM es el más resistente a las nuevas estrategias utilizadas por los ataques de phishing actuales pasando de un



65.62% de precisión a un 88.73% al cabo de cuatro años. También descubrieron que Random Forest es el método recomendado entre todos los métodos evaluados con el nuevo conjunto de datos con una precisión del 94.59% y del 92.47% al clasificar las URL de inicio de sesión.

**Tabla 3.** Tabla resumen de los estudios mencionados en la revisión bibliográfica

**Elaborado por:** Fabiana Jaramillo

Ref	Año	Método	Descripción	Dataset	Técnica	Resultados
[7]	2018	<b>Heurístico</b> Homoglifos Comparación visual	porcentaje de similitud de dos glifos en un nivel de granularidad dado	Lista de glifos que admite una fuente determinada para cada mapa HitZone	mapas HitZone técnica Seekback	porcentaje de similitud = 90% porcentaje de similitud técnica Seekback = 75%
[8]	2020	<b>Heurístico</b> similitud visual	métrica de similitud entre páginas webs iguales a pesar de tener contenidos diferentes	VisualPhish	red neuronal convolucion al (CNN) de tripletes	56% en la coincidencia 30% en el ROC área de clasificación bajo la curva
[9]	2019	<b>Heurístico</b> homógrafo de IDN	Los IDN se extraen con el mismo número de caracteres y ven la similitud de cada letra	SimChar y UC	ShamFinder	detecta ocho veces más homógrafos

[10]	2022	<b>Heurístico</b> Similaridad visual Homoglifos	Similitud en los caracteres (Unicode y ASCII) que aparentan ser idénticas o no pueden distinguirse mediante una inspección visual rápida.	Alexa, PhishTank y PhishStat Dataset generado	Random Forest, y la función hash	Precisión = 99.8%
[11]	2018	<b>Heurístico</b> squatting domains	tipos de dominios ofuscados son: homógrafo, error de tipografía, Bits, Combo, Wrong TLD	PhishTank y Alexa	clasificador Random Forest	exactitud (ACC)= 90% Área bajo la curva (AUC) = 97%
[12]	2020	<b>Heurístico ad hoc</b> Similaridad visual Homoglifos	cada carácter se itera, se asigna a un clúster existente en el que todos los caracteres tienen una similitud con el carácter candidato.	Clustering Homoglyphs into Equivalence Classes	Red neuronal convolucional (CNN)	average precision = 97%

[13]	2021	<b>Basado en Listas</b> Comparación listas negras y blancas	características de Natural Language Processing (NLP) de símbolos en la URL	Phishing Index (PILU-60K)	Support Vector Machines (SVM), Random Forest	SVM precisión = 88,73% , RF precisión = 94,59%
------	------	--	--	---------------------------	--	--

### 1.3. Fundamentación teórica

#### 1.3.1. Seguridad informática

La seguridad informática se encarga de organizar acciones para proteger la confidencialidad, integridad y disponibilidad de la información almacenada en un sistema informático [14]. En otras palabras, plantea políticas de mitigación ante vulnerabilidades existentes en un sistema informático.

#### 1.3.2. Delito informático

Los delitos informáticos son acciones ilícitas cometidas por ciberdelincuentes. El uso indebido de dispositivos tecnológicos y de comunicación, afectan a la intimidad e integridad de un número considerable de personas de una población en común. [15]

#### 1.3.3. Ingeniería Social

Consiste en la manipulación psicológica de las personas para poder obtener información sensible, que de igual manera va de la mano con el Phishing [16]. La Ingeniería Social explota vulnerabilidades humanas, con la búsqueda de obtener la confianza suficiente para que el victimario no sospeche que va a ser engañado. La efectividad de estos tipos de ataque se basa en los siguientes principios:

- **Principio de simpatía**

Es cuando un usuario baja sus defensas ante una persona que gana su confianza o esta alineada a sus intereses. Se dejan seducir por aquellos que a la final buscan el mínimo descuido para sacar provecho de su víctima. [17]

- **Principio de autoridad**

Es coaccionar a las personas a realizar algo fuera de las normas o políticas de seguridad que posean las empresas o instituciones. Engañando al usuario mediante la usurpación de identidad aparentando ser un perfil de un alto directivo de la organización.

- **Principio de escasez o urgencia**

Es un tipo de ataque que engancha a la víctima por medio de la urgencia. Ataques usados comúnmente para filtrar malwares. Por ejemplo, un ransomware que cifra archivos del equipo robando información para pedir un rescate al usuario. [17]

- **Principio de reciprocidad**

Un instinto social muy arraigado a la naturaleza humana es la reciprocidad ante acciones [17]. Este engaño logra manipular al usuario mediante una oferta atractiva. Donde, la persona deberá retribuir al interlocutor ya sea con dinero, servicio, trabajo o un favor.

- **Principio de aceptación social**

La conducta humana como seres sociales, busca tener la aprobación colectiva, invadiendo al juicio racional del cerebro para llevar a cabo acciones no permitidas o indebidas sin pensar en las consecuencias.

#### **1.3.4. Phishing**

Es la combinación de Ingeniería Social y exploits, enfocados al engaño de la víctima solicitando información personal, comúnmente efectuado para conseguir una ganancia monetaria [18]. La mayoría de los ataques de Phishing inciden con el envío de correos

electrónicos falsos, que contienen enlaces o archivos maliciosos camuflados como legítimos sin limitarse a la ingenuidad del factor humano.

### **1.3.5. Phisher**

Persona que hace el Phishing con el fin de estafar a la víctima extorsionándolo y beneficiándose con una retribución económica [18]. El atacante oculta su identidad mediante el robo de la misma pretendiendo ser una persona, corporación o servicio de confianza. El atacante engaña al usuario para que realice acciones que llegan a dañar a la víctima.

Existen diferentes tipos de ataques de Phishing:

- **Spear Phishing**

Es un ataque dirigido a personas o empresas específicas en lugar de usuarios al azar. Es una versión más profunda del Phishing que requiere un conocimiento especial sobre una organización, incluida su estructura de poder. En este ataque, los correos electrónicos se envían a personas concretas, a diferencia del hishing. [19]

- **Whaling**

Se conoce como ataque Whaling Phishing. Es una forma de spear Phishing donde, en este Phishing, los atacantes se dirigen a empleados de alto perfil, como el director ejecutivo o el director financiero, para robar información confidencial de una empresa. Como estas personas ocupan puestos más altos dentro de la empresa, tendrán acceso completo a datos confidenciales. Será fácil obtener más información. [20]

- **Smishing**

También se conoce como Phishing por Servicio de Mensajes Cortos (SMS). Es un tipo de ataque de Ingeniería Social llevado a cabo para robar datos de los usuarios, incluida información personal, información financiera y credenciales. Smishing también tiene como objetivo lavar el dinero de las víctimas. En Smishing, los estafadores envían mensajes de Phishing a través de un mensaje de texto SMS que incluye un enlace malicioso. Los mensajes de Phishing engañan a los destinatarios para que hagan clic

en el enlace malicioso, que los redirige a una página de Phishing donde se encuentra la información personal. [20]

- **Vishing**

También se conoce como Phishing de voz. Es un tipo de fraude telefónico que utiliza mensajes de voz para obtener información personal o dinero de las víctimas. Vishing utiliza grabaciones de voz automatizadas para atraer a las víctimas. En Vishing, se envía una llamada de voz automatizada que indica que la cuenta bancaria de los destinatarios se ha visto comprometida. Luego, el mensaje de voz le pide al destinatario que llame a un número gratuito específico. Una vez que los usuarios llaman a ese número gratuito, el número de cuenta bancaria del usuario y otros datos personales se recopilan a través del teclado del teléfono. [20]

- **Pharming**

El Pharming a veces se conoce como Phishing sin señuelo. Cuando un usuario intenta navegar a un sitio, su computadora puede determinar la dirección IP consultando un archivo local de asignaciones definidas (un archivo de hosts) o consultando un servidor de sistema de nombres de dominio (DNS) en Internet. El Pharming generalmente se lleva a cabo cambiando el archivo de hosts en la computadora de la víctima (Pharming de archivos de hosts) o explotando una vulnerabilidad en el software del servidor DNS (envenenamiento de DNS). [20]

- **Content-injection Phishing**

En esto, el contenido del sitio web legítimo se reemplaza con algún contenido aleatorio con diferentes campos de entrada similares al sitio legítimo. Los usuarios finales confían y proporcionan sus datos fácilmente. [20]

- **Search engine Phishing**

Ocurre cuando los phishers crean sitios web con ofertas que suenan atractivas y los indexan legítimamente con los motores de búsqueda. Los usuarios encuentran estos sitios en el curso normal de la búsqueda de productos o servicios y son engañados para que brinden su información. [20]

### **1.3.6. Mecanismos de seguridad**

Los Mecanismos de seguridad son aquellos mecanismos que están diseñados para detectar, alertar o recuperarse de un ataque de seguridad. Ejecutan varios servicios básicos de seguridad además de especificar como deben ser efectuados los controles. [16]

Se clasifican en tres grupos:

- **Prevención**

Se trata de una medida que se toma de forma precavida, con el fin de evitar situaciones negativas que incumplan con las normas de seguridad de una empresa o institución.

- **Detección**

Se basa en la detección de actividades inapropiadas, indebidas o irregulares que ocasionan infracciones o intenciones de quebrantamientos en la seguridad de un sistema, logrando arriesgar a la empresa o institución.

- **Recuperación**

Conjunto de técnicas y procedimientos para acceder y extraer información almacenada de distintos dispositivos o servidores. Que por mal e infortunio no se logra acceder de modo habitual. El objetivo es recobrar la normalidad de las actividades dentro de la empresa o institución.

### **1.3.7. Integridad de los datos**

Se encarga de la validez y coherencia de la información sin que haya sido modificada o duplicada [14]. De esta manera, se puede descubrir si se adicionado, alterado o eliminado algún tipo de dato.

### **1.3.8. Suplantación de identidad**

Desde el punto de vista informático, se trata de la apropiación ilícita de la identidad de otra persona para actuar en su nombre. Atacantes aprovechan el nivel de confianza que se genera en otras personas para diversos fines de ataques al ciberespacio.

### **1.3.9. Machine learning**

Es un campo de la inteligencia artificial con la capacidad de hacer sistemas avanzados para identificar patrones entre los datos y ser capaces de hacer predicciones.

## **1.4. Objetivos**

### **1.4.1. Objetivo general**

Desarrollar un modelo de Machine Learning para mitigar los fraudes informáticos de phishing basados en la Ingeniería Social en la FISEI.

### **1.4.2. Objetivos específicos**

- Identificar el nivel de conocimiento de la comunidad de la FISEI sobre la detección de los diferentes tipos de ataques Phishing.
- Investigar sobre los algoritmos de Machine Learning para el tratamiento de detección de Phishing en las páginas web que usa la comunidad de la FISEI.
- Implementar un modelo Machine Learning en una Api para generar una extensión de detección de Ingeniería Social tipo Phishing al navegador.



## CAPÍTULO II.- METODOLOGÍA

### 1.1. Materiales

Para el presente proyecto se realizó una encuesta a la comunidad de la FISEI una vez identificado a cada estrato: estudiantes, personal administrativo, personal docente y personal de servicio.

#### Encuesta realizada a la comunidad de la FISEI

La encuesta tiene como objetivo obtener información acerca del nivel de conocimiento sobre la detección del Phishing y conocer la realidad sobre el tratamiento y accionar del personal administrativo, personal docente, personal de servicio y estudiantes ante el delito informático tipo Phishing. La confiabilidad y validez de la encuesta fue validada mediante el coeficiente  $\alpha$  (alpha) de Cronbach el cual dio como resultado un valor de 80% que indica un nivel alto de fiabilidad. (Ver **Anexo A**)

Las opciones de respuesta consisten en una escala de Likert para calificar el nivel de respuestas de los encuestados, las opciones de cada ítem se muestran en la **Tabla 6**:

**Tabla 4.** Encuesta a la comunidad FISEI

**Elaborado por:** Fabiana Jaramillo

N°	Ítem				
1	¿Su grado de conocimiento sobre el delito informático asociado al robo de identidad a través de la internet es?				
	Muy Alto	Alto	Medio	Bajo	Ninguno
2	¿Se encuentra propenso a algún tipo de ataque de Ingeniería Social que comprometa su información confidencial?				
	1	2	3	4	5
3	¿El robo de identidad se puede prevenir con algún tipo de herramienta informática?				
	Totalmente de acuerdo	De acuerdo	Indeciso	En desacuerdo	Totalmente en desacuerdo
4	¿Conoce sobre el delito informático Phishing?				
	1	2	3	4	5
5	¿Ha escuchado en el medio si alguien ha sido víctima de un ataque de Phishing (robo de identidad)?				

	1	2	3	4	5
6	¿Tiene conocimiento para identificar sitios dudosos que impacten en el robo de identidad?				
	1	2	3	4	5
7	¿Considera que el desconocimiento de los delitos informáticos compromete la seguridad de la información desembocando en el robo de identidad?				
	Totalmente de acuerdo	De acuerdo	Indeciso	En desacuerdo	Totalmente en desacuerdo
8	Califique la importancia de la información que maneja, en caso de que su equipo se vea comprometido a algún delito informático				
	Muy importante	Importante	Moderadamente importante	De poca importancia	Sin importancia
9	¿Utiliza algún tipo de seguridad al navegar en la internet para prevenir ser víctima de robo de identidad?				
	Muy frecuentemente	Frecuentemente	Ocasionalmente	Raramente	Nunca
10	¿Estaría dispuesto a utilizar una herramienta informática en el navegador para aumentar la seguridad?				
	Totalmente de acuerdo	De acuerdo	Indeciso	En desacuerdo	Totalmente en desacuerdo
11	¿Qué hace usted cuando se le presenta un link (enlace) en un correo electrónico?				
	Verificar la url con la página oficial y lo abre		Acceder porque no entiende el aviso		No presta atención

## 1.2. Métodos

### 1.2.1. Modalidad de la investigación

Las modalidades tomadas en cuenta para la presente investigación son las siguientes:

**Modalidad Bibliográfica** – porque se ha tomado información de libros, artículos, páginas Web y tesis de grado.

**Modalidad Experimental** – se ha considerado la relación de la variable independiente modelo de Machine Learning y su influencia en relación con la variable dependiente Phishing para considerar sus causas y efectos.

**Modalidad de Campo** – porque el estudio del problema es en el lugar donde se están generando los hechos; de esta manera se puede conocer y mejorar los inconvenientes

que se producen en la facultad al no detectar ataques informáticos tipo Phishing. Una gran ventaja para establecer parámetros de solución y así cumplir con los objetivos planteados.

### 1.2.2. Población y muestra

La población considerada para este proyecto será de la comunidad de la FISEI, dividida en *estudiantes*, *personal administrativo*, *personal docente* y *personal de servicio* según el régimen laboral al que pertenecen.

El muestreo de los **estudiantes** será aleatoria estratificada proporcional con estratos por niveles de primero a tercero, de cuarto a sexto y de séptimo a noveno semestre, debido a que el conocimiento de los estudiantes no es el mismo en cada nivel.

Para el cálculo del tamaño de la muestra, se utilizó la siguiente fórmula:

$$n = \frac{N\sigma^2Z^2}{(N - 1)e^2 + \sigma^2Z^2}$$

Donde:

n = Tamaño de la muestra.

N = Población (de estudiantes 1840 y de personal según el régimen laboral 100)

Z = Nivel de confianza 95% equivalente a 1,96.

$\sigma$  = Desviación estándar constante 0,5.

e = Margen de error: 5%.

La población total de estudiantes es de 1840 obteniendo un tamaño de muestra de 318 de la comunidad estudiantil de la FISEI.

Para la asignación del tamaño de la muestra de cada estrato, se utilizó la siguiente fórmula:

$$n_i = n \left( \frac{N_i}{N} \right)$$

Donde:

$n_i$  = Tamaño de la muestra de cada estrato.

$n$  = Tamaño de la muestra (de estudiantes 318 y de personal según el régimen laboral 80)

$N_i$  = Número de unidades muestrales en el estrato “ $i$ ”.

$N$  = Número de unidades muestrales en la población (de estudiantes 1840 y de personal según el régimen laboral 100)

$L$  = Número de estratos 3.

**Tabla 5.** Población y Muestra de la Comunidad Estudiantil

**Elaborado por:** Fabiana Jaramillo

<b>Comunidad Estudiantil</b>	<b>Población</b>	<b>Muestra</b>
Primer a Tercero	744	128
Cuarto a Sexto	704	122
Séptimo a Noveno	392	68
	<b>Total</b>	318

La población total del **personal administrativo, personal docente y personal de servicio** según el régimen laboral al que pertenecen son 100, se trabajará con un tamaño de muestra de 80.

**Tabla 6.** Población y Muestra del Régimen Laboral

**Elaborado por:** Fabiana Jaramillo

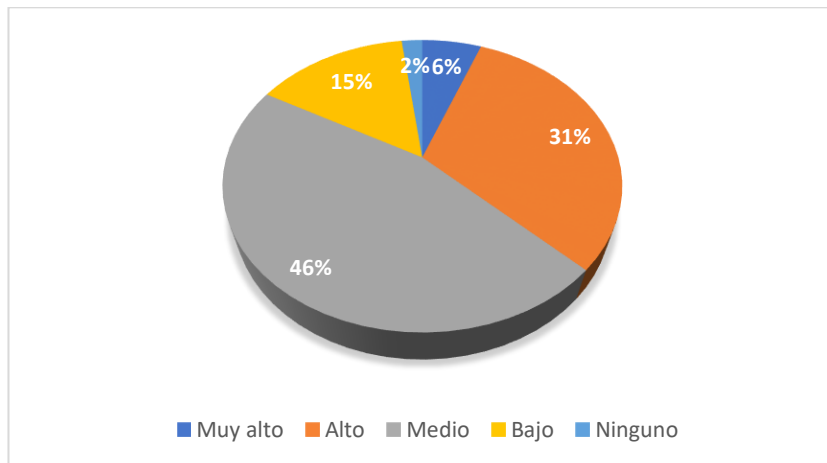
<b>Régimen Laboral</b>		<b>Población</b>	<b>Muestra</b>
<b>Servicio civil público</b>	Personal administrativo	23	18
<b>Otros regímenes especiales</b>	Personal Docente	70	56
<b>Código de Trabajo</b>	Personal de Servicio	7	6
		<b>Total</b>	80

### 1.2.3. Recolección de la información

Tras haberse aplicado las encuestas se obtuvieron los siguientes resultados.

#### Resultado de la encuesta aplicada a la comunidad de la FISEI

- **Ítem 1:** ¿Su grado de conocimiento sobre el delito informático asociado al robo de identidad a través de la internet es?



**Figura 1.** Conocimiento sobre el robo de identidad a través de la internet

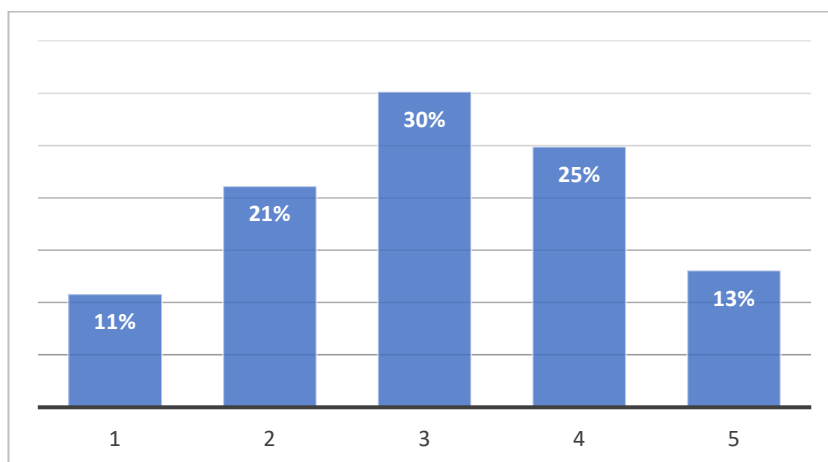
**Elaborado por:** Fabiana Jaramillo

#### Análisis e interpretación de resultados:

De acuerdo con los resultados representados en la **Figura 1**, se puede apreciar varios aspectos como son:

- Un 46% que dice tener un conocimiento medio, junto con un 37% que señala tener conocimiento del tema, lo que puede poner de manifiesto una pequeña apatía de las personas por seguir actualizándose sobre los robos de identidad a través de la internet y prevenir ser víctimas de ciberdelincuentes.
- Un 15% reconoce tener un conocimiento bajo sobre el robo de identidad a través de la internet y un 2% ningún conocimiento del tema, que aunque puede no ser alarmante, no hay que perder de vista ese desconocimiento frente a un delito informático, ya que toda brecha que vean los atacantes les es beneficioso para hacer actos maliciosos.

- **Ítem 2:** ¿Se encuentra propenso a algún tipo de ataque de Ingeniería Social que comprometa su información confidencial?



**Figura 2.** Perspectiva ante la posibilidad de ser víctima de un ataque de Ingeniería Social

**Elaborado por:** Fabiana Jaramillo

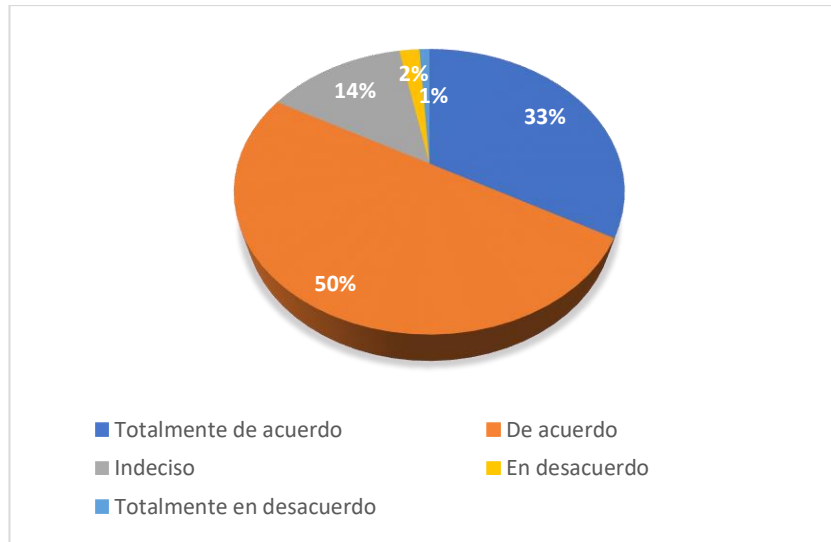
#### **Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 2**, se puede apreciar lo siguiente:

- El 30% de los encuestados no descarta la posibilidad de ser víctima de un ataque de Ingeniería Social, un porcentaje optimista ante el resultado de la primera pregunta, de un 46% que dice tener un nivel medio de conocimiento sobre robos de identidad a través de la internet.
- El índice de ser propensos a algún tipo de ataque de Ingeniería Social varía entre un 13% y un 25% de los encuestados, que aunque no sea preocupante, el peligro aún sigue presente y si en aquellos porcentajes existe la mínima posibilidad de atacar, los ciberdelincuentes la aprovecharan al máximo.
- El 32% de los encuestados señalan que no existe o es muy poco probable que sean víctimas de un ataque de Ingeniería Social, en comparación con un 30% que no

descarta esa posibilidad. Por lo tanto, demuestran que su nivel de persuasión frente a una situación comprometedoras es medio y su accionar tiende a ser cauteloso.

- **Ítem 3:** ¿El robo de identidad se puede prevenir con algún tipo de herramienta informática?



**Figura 3.** Herramienta informática frente a la prevención del robo de identidad

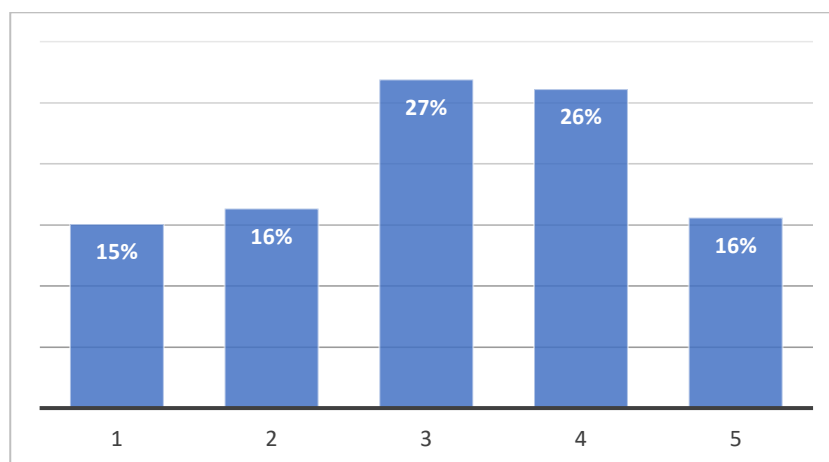
**Elaborado por:** Fabiana Jaramillo

#### **Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 3**, se puede apreciar varias observaciones como:

- Una gran cantidad de encuestados el 83% está de acuerdo, con que el robo de identidad se puede prevenir con una herramienta informática. Por lo tanto, se evidencia la dependencia que el usuario crea con la tecnología.
- Hay que tener en cuenta que una herramienta informática ayuda a mitigar el problema no a terminarlo. Es por tal motivo que un 14% de los encuestados se encuentra indeciso, junto con un 3% que se encuentra en desacuerdo.

- **Ítem 4:** ¿Conoce sobre el delito informático Phishing?



**Figura 4.** Conocimiento sobre Phishing

**Elaborado por:** Fabiana Jaramillo

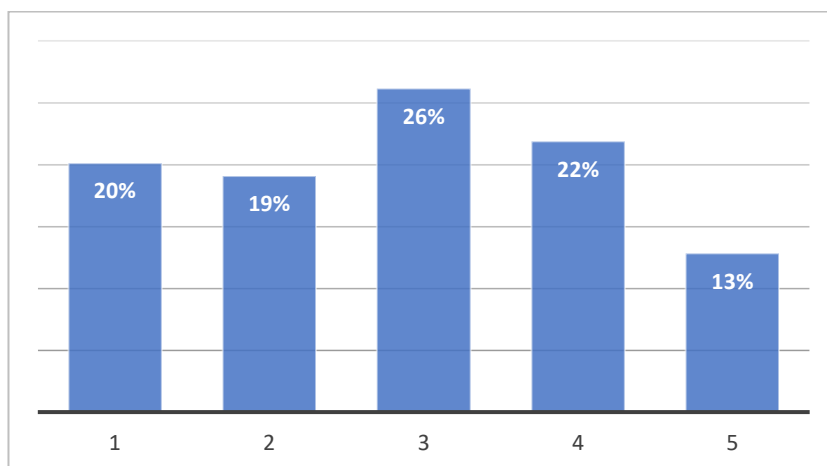
#### **Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 4**, se puede apreciar varios aspectos como son:

- Manifiestan que un 27% tiene un conocimiento medio sobre el Phishing y otro 26% dice conocer del Phishing, y el 16% sostiene que conoce lo suficiente sobre el tema. Comparando estos porcentajes con los del ítem 1 de un 46% de un nivel medio de conocimiento sobre el robo de identidad a través de la internet, con un 31% alto y un 6% muy alto, se puede decir que son porcentajes un poco bajos. Los encuestados en el primer ítem dicen conocer de robos de identidad, pero al mencionarles sobre Phishing los porcentajes de su nivel de conocimiento disminuye.
- Un 16% manifiesta conocer poco sobre el tema y un 15% no sabe nada del tema. De igual manera comparado con un 15% de un conocimiento bajo sobre el robo de identidad a través de la internet y un 2% sin conocer nada del tema, resultados del ítem 1. Se puede observar que no varía mucho el nivel bajo, pero aumenta el porcentaje de ignorancia del tema, demostrando la verdadera realidad sobre el nivel de conocimiento que tienen los encuestados.



- **Ítem 5:** ¿Ha escuchado en el medio si alguien ha sido víctima de un ataque de Phishing (robo de identidad)?



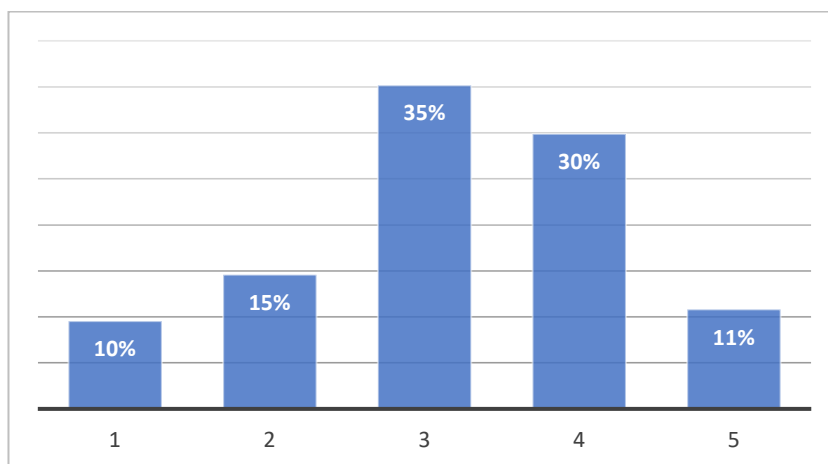
**Figura 5.** Víctimas de Phishing  
**Elaborado por:** Fabiana Jaramillo

**Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 5**, se puede apreciar lo siguiente:

- Un 26% dice haber escuchado ocasionalmente de alguien que ha sido víctima de un ataque de Phishing, un 22% frecuentemente y un 13% mucho. Los resultados son totalmente coherentes con los porcentajes del ítem 2, de un 30% que no descarta la posibilidad de ser víctima de un ataque de Ingeniería Social, junto con un 25% y 13% que señalan ser propensos a un ataque de Ingeniería Social.
- Un 19% dice que ha escuchado poco del tema y un 20% nada, tendencias bajas que llegan a alarmar y afirmar una vez más que el peligro de ser víctima de un ataque de Phishing está presente.

- **Ítem 6:** ¿Tiene conocimiento para identificar sitios dudosos que impacten en el robo de identidad?



**Figura 6.** Conocimiento para identificar sitios dudosos

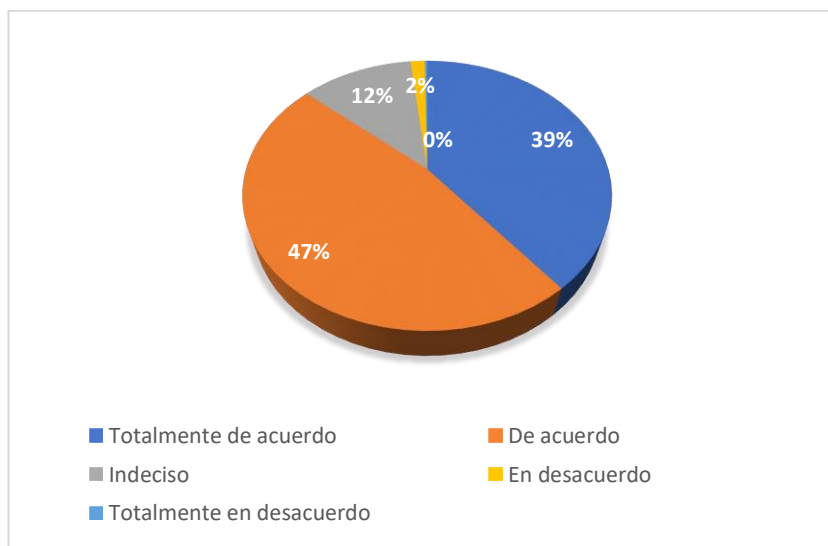
**Elaborado por:** Fabiana Jaramillo

#### **Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 6**, se puede apreciar varios aspectos como son:

- El 35% reconoce que su conocimiento para identificar sitios dudosos es medio, llegando a decir que la posibilidad de ser víctimas potenciales de un ataque Phishing no se descarta. Un 30% señala tener conocimiento y un 11% dice tener el conocimiento suficiente para poder identificar un sitio dudoso. De acuerdo con los porcentajes del ítem 4 sobre el conocimiento de Phishing de un 27% medio, 26% alto y 16% muy alto, se demuestra que los encuestados muestran interés por protegerse y tratar de prevenirse de un ataque informático.
- Hay que resaltar también que un 15% dice conocer muy poco y un 10% nada, que aunque es una tendencia baja, no se puede dejar de lado ni dejar de preocuparse por los que dicen poder identificar un sitio dudoso, ya que, hay varios métodos de ataques que burlan la capacidad de identificación de sitios dudosos que tienen los usuarios.

- **Ítem 7:** ¿Considera que el desconocimiento de los delitos informáticos compromete la seguridad de la información desembocando en el robo de identidad?



**Figura 7.** Desconocimiento de los delitos informáticos

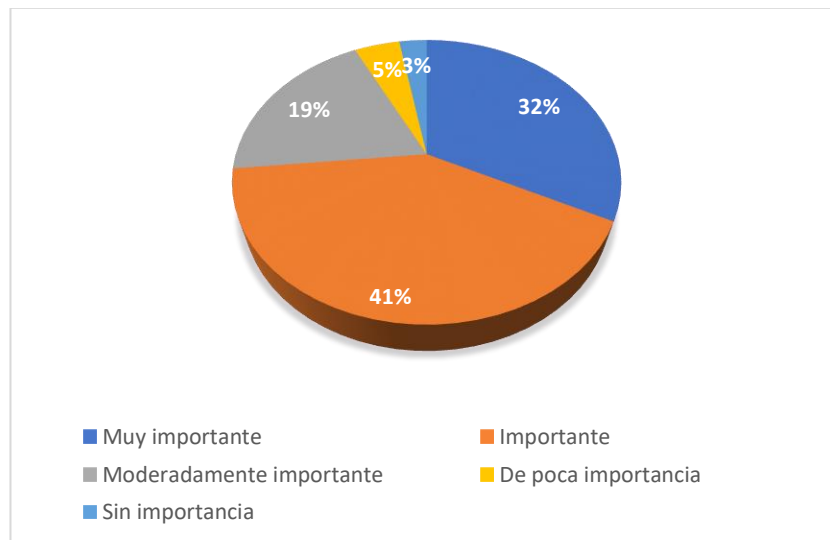
**Elaborado por:** Fabiana Jaramillo

#### **Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 7**, se puede apreciar varias observaciones como:

- Un alto índice de un 39% que están totalmente de acuerdo y un 47% que esta de acuerdo con el ítem 7. El desconocimiento es un problema grave y bastante vulnerable si de delitos se habla; no importa el fraude, el daño que provocan puede ser en cierto modo irremediable.
- Un 12% dice estar indeciso junto con un 2% que esta en desacuerdo, el índice es bajo, pero demuestra en ese pequeño porcentaje una falta de interés por querer conocer y actualizarse. Tal población puede convertirse en una gran ventaja para delincuentes informáticos, innovadores e ingeniosos a la hora de persuadir a su víctima.

- **Ítem 8:** Califique la importancia de la información que maneja, en caso de que su equipo se vea comprometido a algún delito informático



**Figura 8.** Importancia de la información

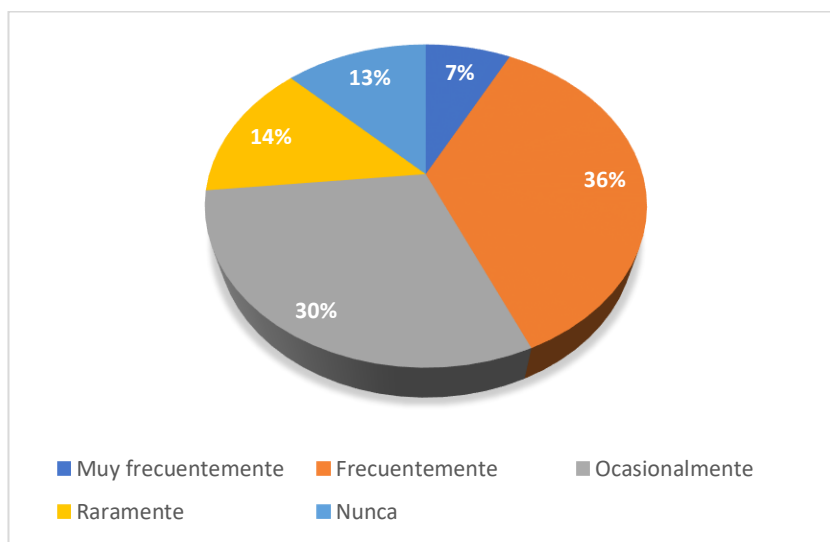
**Elaborado por:** Fabiana Jaramillo

**Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 8**, se puede apreciar lo siguiente:

- De acuerdo con un 19%, de los encuestados que dice que su información es moderadamente importante, con un 41% que la considera importante y un 32% muy importante. Se deduce, que la mayoría de la población maneja información valiosa, personal y confidencial. Si tal información por algún ataque de Phishing llega a comprometerse causaría daños angustiosos a sus propietarios.
- Por otro lado, un 5% señala que la información que maneja es de poca importancia y otro 3% señala que no es de importancia. Un índice bajo, pero con una tendencia a crecer.

- **Ítem 9:** ¿Utiliza algún tipo de seguridad al navegar en la internet para prevenir ser víctima de robo de identidad?



**Figura 9.** Utilización de algún tipo de seguridad al navegar en la internet

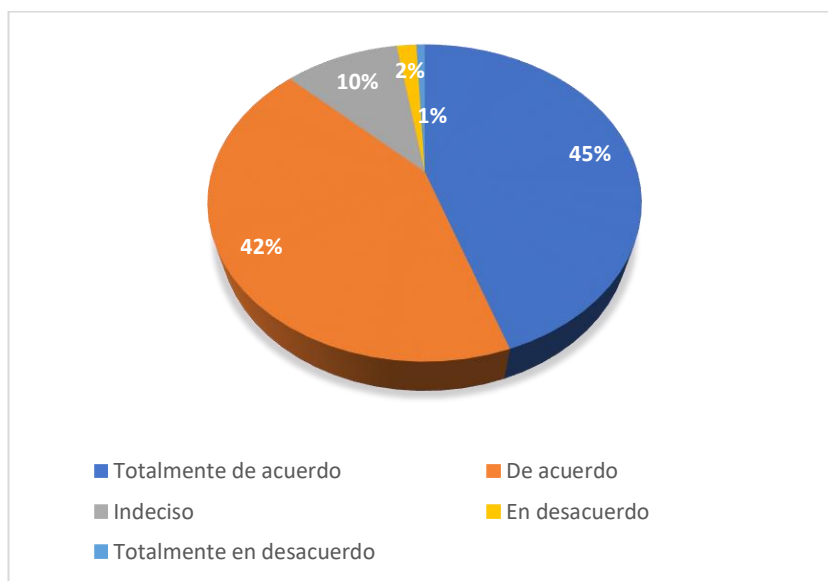
**Elaborado por:** Fabiana Jaramillo

#### **Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 9**, se puede apreciar varios aspectos como son:

- Un 36% que dice usar frecuentemente algún tipo de seguridad y un 7% señala que lo usa muy frecuentemente. El resultado es muy alentador ya que en los porcentajes del ítem 2 la mayoría de los encuestados señalaron estar de acuerdo, que el uso de una herramienta informática puede llegar a prevenir el robo de identidad. Claro está que hay que tener en cuenta que no elimina el ataque, solo lo mitiga.
- El 30% reconoce usar ocasionalmente algún tipo de seguridad, seguido de un 14% que dice usarla raramente y un 13% reconoce que nunca la usó. Son porcentajes que tienen la tendencia a subir, demostrando una vez más el interés que tiene la población en protegerse y no ser víctimas de ciberdelincuentes.

- **Ítem 10:** ¿Estaría dispuesto a utilizar una herramienta informática en el navegador para aumentar la seguridad?



**Figura 10.** Disponibilidad de usar una herramienta informática

**Elaborado por:** Fabiana Jaramillo

#### **Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 10**, se puede apreciar lo siguiente:

- Una herramienta informática llama la atención de la mayoría de la población demostrado en la encuesta realizada ya que un 45% esta totalmente de acuerdo y otro 42% de acuerdo. Índices altos que apoyan y alientan al desarrollo del modelo de Machine Learning para el tratamiento de detección de Phishing.
- El 10% se encuentra indeciso, otro 2% esta en desacuerdo y un 1% esta totalmente en desacuerdo. Los porcentajes son bajos, pero de igual manera que el ítem 9 tienden a subir. Demostrando un grado de aceptación favorable al uso de una herramienta informática para aumentar la seguridad de navegación en la internet.

- **Ítem 11:** ¿Qué hace usted cuando se le presenta un link (enlace) en un correo electrónico?



**Figura 11.** Accionar ante la presencia de un link (enlace) en un correo electrónico

**Elaborado por:** Fabiana Jaramillo

#### **Análisis e interpretación de resultados:**

De acuerdo con los resultados representados en la **Figura 11**, se puede apreciar varios aspectos como son:

- Hay un 43% que al link (enlace) lo verifica con la página oficial y luego lo abre, pero como ya se había mencionado los métodos de ofuscación que usan los ciberdelincuentes logra distraer al usuario y persuadirles en su engaño.
- Un 46% dice no prestar atención a este tipo de correos Phishing, un porcentaje bastante acercado a la realidad pero que de igual manera no hay que dejar de lado, ya que se resalta la astucia con la que actúa un ciberdelincuente cuando quiere conseguir su objetivo.
- Aunque el porcentaje de acceder al enlace porque no se entiende el aviso, es bajo de un 11%, para los ciberdelincuentes es una entrada muy amplia a la vulnerabilidad de ataques de Ingeniería Social tipo Phishing.

#### **1.2.4. Procesamiento y análisis de datos**

De acuerdo con las encuestas aplicadas a estudiantes, personal administrativo, personal docente y personal de servicio según el régimen laboral al que pertenecen. Se determinaron los siguientes aspectos relacionados con el nivel de conocimiento que tiene la comunidad de la FISEI sobre la detección del Phishing y la factibilidad del desarrollo del modelo de Machine Learning para generar una extensión de detección de Phishing al navegador.

- Se considera que la mayoría de la comunidad de la FISEI tiene un grado de conocimiento medio sobre el delito informático Phishing asociado al robo de identidad a través de la internet; el ataque de Phishing (robo de identidad) es muy escuchado en el medio. Por lo tanto, no hay que dejar de preocuparse, aunque exista ese interés de conocer del tema, aún hay personas que lo desconocen, convirtiéndose en un problema grave y bastante vulnerable si de delitos se habla. Los phishers aprovecharán ese desconocimiento para engañar al usuario.
- La comunidad de la FISEI está muy propensa a ataques tipo Phishing que pueden comprometer su información que en general la han calificado como importante.
- Existe un alto nivel de aceptación sobre el uso de una herramienta informática para prevenir ser víctima de ataques de Phishing (robo de identidad) y aumentar la seguridad en el navegador.
- Aunque la tendencia de acceder al enlace verificando la URL es alta cuando se presenta en un correo electrónico, la existencia de métodos de ataques burla la capacidad de reconocimiento visual que tienen los usuarios al comprobar la veracidad de la URL.



## CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

### 3.1. Análisis y discusión de los resultados.

Luego de haber realizado las encuestas para identificar el nivel de conocimiento que la comunidad de la FISEI tiene sobre la detección del delito informático Phishing. Se procederá a realizar experimentaciones mediante un hackeo ético en donde se simulará un ataque Phishing dirigido a la comunidad de la FISEI. Después, se realizará análisis y discusiones de: algoritmos de Machine Learning a utilizar, metodologías de cada uno de los procesos de un modelo de Machine Learning y, por último, metodologías para generar una extensión al navegador.

#### 3.1.1. Experimentación simulación de ataque Phishing

En este trabajo se propone realizar una simulación de ataque mediante una clonación de páginas. Como primer paso se realizó una pequeña encuesta, con el fin de obtener un ranking de las páginas más visitadas por la comunidad de la FISEI.

- Liste las páginas web más visitadas por usted



**Figura 12.** Ranking de las páginas web más visitadas

**Elaborado por:** Fabiana Jaramillo

- Liste las páginas web más visitadas por usted, estando en la facultad FISEI



**Figura 13.** Ranking de las páginas web más visitadas en la FISEI

**Elaborado por:** Fabiana Jaramillo

- Liste las redes sociales que más utiliza



**Figura 14.** Ranking de las redes sociales más utilizadas

**Elaborado por:** Fabiana Jaramillo

De acuerdo con los resultados de los rankings de las páginas más visitadas se clonaron las siguientes páginas: la plataforma educativa de la FISEI, el sistema integrado de la UTA, facebook y netflix. Para las páginas de youtube y google que en la **Figura 12.** se muestran como las más visitadas, se envió un formulario que solicita información personal a la víctima. La técnica que se usó para la ejecución de esta simulación de ataque Phishing fue la aplicación de Ingeniería Social en los correos que se enviaron a las víctimas.

Para la metodología de clonación de páginas se utilizaron varias herramientas como: La aplicación Htrack Website Copier que captura el sitio web y lo descarga. El proyecto descargado se lo edita por medio del id de desarrollo Visual Studio Code. Cada uno de los proyectos son subidos al repositorio Github. El repositorio GitHub permite personalizar el dominio de la página. Los dominios son creados específicamente para engañar al usuario, usando técnicas de ofuscación como: *Domain Name Squatting*. Las diferentes paginas clonadas disponen de un contador de visitas, su respectivo login y un formulario de actualización de datos. Los datos son información que un Phisher trata de conseguir cuando realiza un ataque Phishing. La información del contador de visitas y los datos capturados del login y de cada formulario se almacenan en una base de datos. El fin de esta experimentación es poder ver el alcance del usuario ante un ataque Phishing.

**Inconvenientes:** Al subir los proyectos de las páginas clonadas al repositorio GitHub y otorgarle un dominio personalizado, claro está ofuscándolo del dominio original. El repositorio suspendió el uso de la cuenta. Por lo que se concluye que el uso de este repositorio tiene términos de uso estrictos con este tipo de metodologías de ataques Phishing.

Dicho esto, se cambió la subida y publicación de los proyectos a un hosting gratuito llamado TonoHost. La **Figura 15.** muestra el esquema de funcionamiento de la simulación del ataque Phishing.

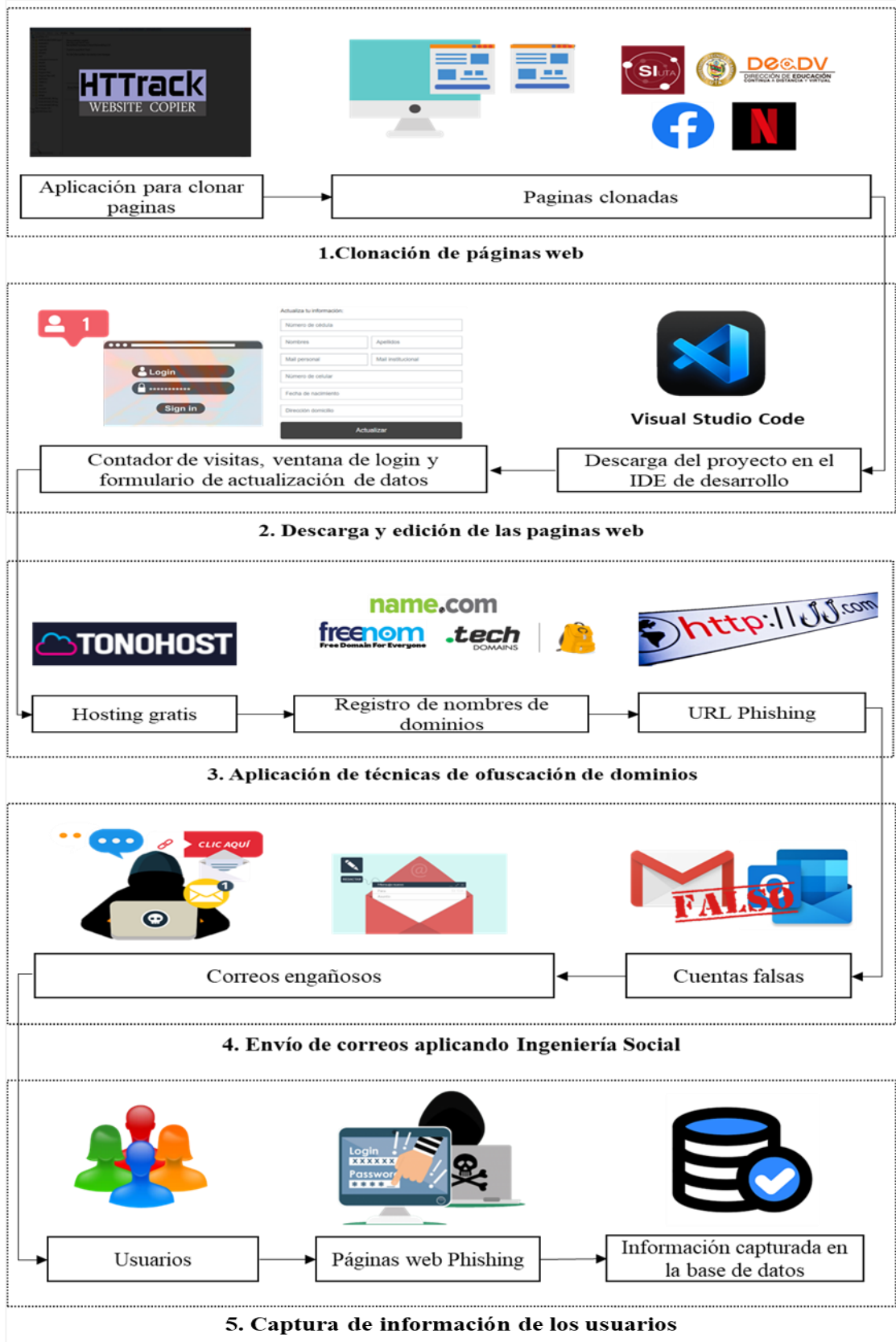


Figura 15. Esquema de simulación de ataque Phishing

Elaborado por: Fabiana Jaramillo

## **Respaldo legal**

Para encasillar a la simulación de ataque Phishing dentro de los parámetros de legalidad o ilegalidad hay que tener claro dos conceptos primordiales: delito y hacking ético.

Según la legislación ecuatoriana un delito es una acción antijurídica, culpable o dolosa que es sancionada con una pena [21]. Ahora bien, un hackeo ético se lo puede definir como el análisis de vulnerabilidades con el único propósito de favorecer y prevenir ataques malintencionados.

Una vez aclarada la definición de delito y hackeo ético; la simulación de ataque Phishing para este estudio hace parte de un hackeo ético y no se encuentra penado por la ley ecuatoriana, la cual es muy relevante en cuanto a intrusiones y robo de información usando herramientas como spamming, Pishing, tampering, entre muchas otras, sancionando así el hecho de robar información, estafar al cliente y romper seguridades causando perjuicio a la víctima.

En definitiva, la constitución ecuatoriana no contempla la posibilidad de que alguien haga uso de herramientas diseñadas en gran porcentaje, para explotar las vulnerabilidades, en beneficio de los sistemas de destino, transformando las mismas herramientas en parte de la solución y no del problema.

## **Compromiso Ético**

El presente trabajo de integración curricular documenta el siguiente compromiso: Toda la información recolectada será usada con fines estrictamente académicos, será cuidadosamente custodiada, anonimizada dentro del proyecto de investigación. Los datos originales se destruirán una vez procesados.

### **3.1.2. Métodos de ofuscación de dominios**

El término de ofuscación de dominios se lo puede definir como: Técnicas de enmascaramiento usadas por los atacantes como medidas de burlar el análisis estático

realizado por medio de las propiedades léxicas a la URL, o, para eludir los mecanismos basados en listas negras.[22]

Los phishers tienen acceso cada vez más grande a un montón de métodos para ofuscar el destino final de la web. Los ataques de Phishing en su mayoría redirigen al usuario a una página web clonada, con el fin de tener acceso a credenciales o a información sensible de la víctima.

Los métodos más comunes de la ofuscación se muestran en la **Tabla 7**.

**Tabla 7.** Métodos de ofuscación de dominios

**Elaborado por:** Fabiana Jaramillo

<b>Métodos</b>	<b>Características</b>
Ingeniería Social	<ul style="list-style-type: none"> <li>- Ataque clásico.</li> <li>- Modificar por HTML la dirección a la que apunta el enlace.</li> </ul>
Redirecciones	<ul style="list-style-type: none"> <li>- Con un script forzara a que al visitar alguna de las páginas del dominio real, este redirija a otro.</li> <li>- Uso de inyecciones de código.</li> </ul>
Open-Redirect en Diversos Servicios	<ul style="list-style-type: none"> <li>- Aprovecha vulnerabilidades que permiten realizar redirecciones con las APIs de compartir que utilicen Open-Redirect.</li> <li>- Posibilidad de cambiar el texto, descripción e imagen en las URLs compartidas en Facebook.</li> </ul>
Domain Name Squatting	<ul style="list-style-type: none"> <li>- Registrar nombres de dominios que puedan causar confusión con marcas reconocidas.</li> <li>- Da lugar a la suplantación de sitios webs originales.</li> </ul>

Acortadores de Enlaces	<ul style="list-style-type: none"> <li>- Reducir las dimensiones de las URLs</li> <li>- El usuario no sabe hacia donde apunta el enlace.</li> </ul>
Navegadores Empotrados en las Apps Móviles	<ul style="list-style-type: none"> <li>- Las redes sociales, abren enlaces externos en un navegador capado que carga dentro de la propia aplicación.</li> <li>- El usuario es incapaz de ver la URL a la que está accediendo por el tiempo de procrastinación en las redes sociales.</li> </ul>

Realizado un análisis a la **Tabla 7**, el enfoque del presente estudio se enmarca en el método de ofuscación por medio de Ingeniería Social y Domain Name Squatting. Los métodos logran burlar y confundir al usuario con nombres de dominios que a simple vista no se puede llegar a identificar algún error.

En el método de Domain Name Squatting se incluyen distintas técnicas como:

- **Homógrafos**

El squatting basado en homógrafos se refiere al squatting que se parecen a los dominios objetivos en la percepción visual. Por ejemplo, dos caracteres “rn” se puede suplantar con el carácter “m”. faceb00k es un homógrafo squatting a facebook ya que "00" se parece a "oo". En esta técnica sacan provecho de dominios internacionalizados (IDN), utilizan la codificación Punycode para convertir caracteres Unicode en ASCII; de esta manera la URL pasa desapercibida en el navegador.[11]

- **Typo squatting**

El objetivo de esta técnica es imitar los nombres de dominio mal escritos por los usuarios. Existen varios métodos para generar typo squatting basado en un dominio objetivo dado, incluyendo inserción (adición de un carácter), omisión (eliminación de un carácter) repetición (duplicación de un carácter) e intercambio de vocales (reordenación de dos caracteres consecutivos). La inserción se refiere a añadir un carácter adicional al dominio original. La omisión se refiere a la supresión de un

carácter en el dominio. La repetición se refiere a la repetición de un carácter en el dominio. El intercambio de vocales consiste en la reordenación de dos caracteres consecutivos en el dominio. [11] Por ejemplo, facebook.tech es un dominio de inserción al adicionar la “c”.

- **Bits**

El squatting de bits consiste en dar la vuelta a un bit del nombre de dominio. Un dominio de squatting de bits es sólo un bit diferente del dominio de destino objetivo [11]. Por ejemplo, youtube.tk es un dominio con squatting de bits donde un bit "u" se cambia por "o".

- **Combo**

El squatting de combo consiste en concatenar el nombre del dominio de destino con otros caracteres. La concatenación puede estar unida al principio o al final. Para este trabajo se ha utilizado el combo squatting con guiones que están permitidos en el nombre de dominio. [11] Por ejemplo, servicios-uta-edu-ec es el combo squatting en el que se adjuntan nuevos caracteres con un guión.

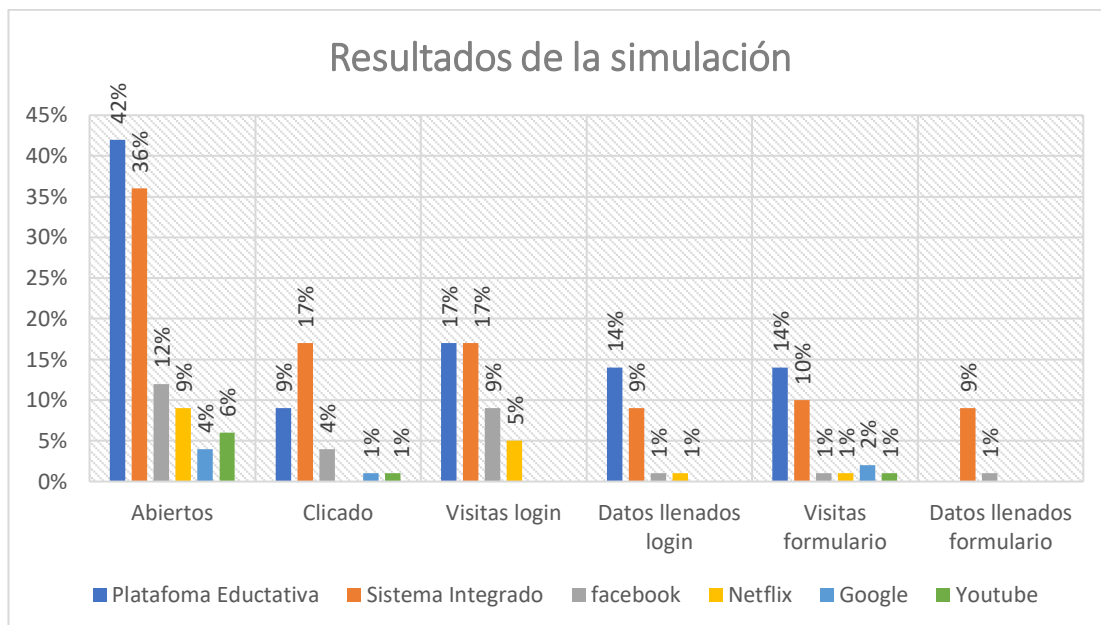
- **WrongTLD**

Todas las técnicas de squatting anteriores se centran en el nombre de dominio, pero ignoran el dominio de nivel superior o en inglés top-level domain, un TLD. WrongTLD se refiere a dominios que cambian el TLD pero mantienen el nombre de dominio como el mismo. [11] Por ejemplo, netflix.cf pertenece a la categoría de TLD erróneo, ya que el TLD original “com” se cambia a “cf”.

### **3.1.3. Resultados de la simulación de ataque Phishing**

Una vez realizado la simulación de ataque de Phishing con la clonación de siguientes páginas: la plataforma educativa de la FISEI, el sistema integrado de la UTA, facebook, netflix y los formularios enviados como un engaño de parte de google y youtube (**Anexo B**) se obtienen los siguientes resultados.





**Figura 16.** Resultados de la simulación del ataque Phishing

**Elaborado por:** Fabiana Jaramillo

En general, la plataforma educativa y el sistema integrado tienen un mejor desempeño en términos de tasa de apertura de correo electrónico y llenado de datos en comparación con otras plataformas como facebook, netflix, google y youtube. Aproximadamente el 42% y 36% de los correos enviados a la plataforma educativa y el sistema integrado, respectivamente, fueron abiertos. Además, aproximadamente el 14% y 9% de los correos enviados a la plataforma educativa y el sistema integrado, respectivamente, resultaron en datos llenados.

Sin embargo, la tasa de clics, visitas al login y al formulario es más baja en la plataforma educativa y el sistema integrado. Aproximadamente el 9% y 17% de los correos enviados de la plataforma educativa y el sistema integrado, respectivamente, resultaron en clics. La tasa de visitas al formulario también fue baja, con aproximadamente el 14% y 10% de los correos enviados de la plataforma educativa y el sistema integrado, respectivamente.

En general, las otras plataformas tienen tasas mucho más bajas de correos abiertos, clics, visitas al formulario y al login en comparación con la plataforma educativa y el sistema integrado.

Después de todo se puede determinar que en la facultad aún hay usuarios que caen en este tipo de ataques. Lo que indica la importancia de este trabajo para la detección de Phishing al implementar una extensión al navegador.

### **3.2. Desarrollo de la propuesta**

Para el desarrollo del proyecto se analizará los lenguajes de desarrollo de Machine Learning, las metodologías de desarrollo, el modelo de Machine Learning y el desarrollo de la extensión al navegador.

#### **3.2.1. Lenguaje de desarrollo de Machine Learning**

Hay diferentes lenguajes de programación que se emplean para desarrollar aplicaciones de Machine Learning. Cada aplicación tiene necesidades y limitaciones específicas, y algunos lenguajes pueden ser más adecuados que otros para abordar ciertos problemas. Es por eso, que se realizó una tabla comparativa entre el lenguaje de programación Python y Matlab para analizar el lenguaje de desarrollo que mejor se ajusta al estudio.

- **Python**

Python es un lenguaje de programación interpretado, muy popular y de uso general. Cuenta con una amplia gama de librerías y módulos que facilitan el desarrollo y la implementación de modelos de aprendizaje automático. Además, Python es un lenguaje de programación versátil y fácil de aprender, lo que lo hace ideal para desarrolladores de todos los niveles. [23]

- **Matlab**

Matlab es un software que está diseñado específicamente para abordar problemas en el campo de la ciencia y la ingeniería. Matlab cuenta con herramientas incorporadas para el análisis de datos, sistema de control, procesamiento de imágenes y procesamiento de señales. Además, Matlab utiliza un lenguaje de programación basado en matrices, lo que hace que sea la forma más apropiada de expresar cálculos matemáticos computacionales. [23]

**Tabla 8.** Tabla comparativa de lenguajes de desarrollo de Machine Learning  
**Elaborado por:** Fabiana Jaramillo

	<b>Lenguajes</b>	
	<b>Python</b>	<b>Matlab</b>
<b>Características</b>	<ul style="list-style-type: none"> <li>- Favorece a aplicaciones de desarrollo web</li> <li>- Favorece a la automatización de procesos</li> </ul>	<ul style="list-style-type: none"> <li>- Ofrece soporte para para scripting</li> <li>- Ofrece soporte para programación procedimental y orientada a objetos [24]</li> </ul>
<b>Nivel de aprendizaje</b>	<ul style="list-style-type: none"> <li>- Es un lenguaje fácil y se puede manejar en distintas tareas de programación [24]</li> </ul>	<ul style="list-style-type: none"> <li>- Es un lenguaje enfocado en la ingeniería y la ciencia, por su matemática matricial [24]</li> </ul>
<b>Librerías y módulos disponibles</b>	<ul style="list-style-type: none"> <li>- Múltiple gama de bibliotecas y módulos que ayudan a la codificación y velocidad en el desarrollo [23]</li> </ul>	<ul style="list-style-type: none"> <li>- Una extensa biblioteca de herramientas integradas (Toolboxes) [23]</li> </ul>
<b>Coste</b>	<ul style="list-style-type: none"> <li>- Gratis y de código abierto</li> </ul>	<ul style="list-style-type: none"> <li>- No es gratis</li> </ul>

Después del análisis comparativo de la **Tabla 8.** se seleccionó Python como el lenguaje de programación del trabajo. Primero por ser un lenguaje de programación más versátil y tener una amplia variedad de librerías y módulos disponibles para abordar diferentes tareas, incluyendo la división de una URL. Además, Python es un lenguaje de código abierto y gratuito, lo que significa que no hay costos asociados con su uso o instalación.

### **3.2.2. Metodología de desarrollo**

A continuación, se realizó una tabla comparativa entre tres metodologías ágiles, para determinar la metodología que mejor se ajuste para el desarrollo del proyecto. La metodología ágil por utilizar debe facilitar el tiempo de desarrollo, garantizar la eficiencia de los recursos, controlar y manejar los procesos adecuadamente de cambios que se presenten durante el desarrollo.

- **SCRUM**

La metodología SCRUM se centra en iteraciones con el objetivo de lograr la máxima predictibilidad en el progreso del proyecto. Con este enfoque, se asume y se controlan los riesgos mientras se entregan los resultados esperados. [25]

- **KANBAN**

La metodología KANBAN se enfoca en la gestión general de la forma en que se llevan a cabo las tareas del proyecto. Esto se hace a través de una tabla que representa los estados de las actividades y su progreso, y que se mueven de acuerdo a un flujo de trabajo previamente establecido. [25]

- **XP**

La metodología XP (Extreme Programming) se basa en un grupo de reglas ampliamente utilizadas en el desarrollo de software, que se aplican de manera flexible para crear un proceso de trabajo ágil. XP enfatiza las tareas que tienen más valor y descarta aquellas que puedan afectar negativamente el proyecto. [25]

**Tabla 9.** Tabla comparativa de metodologías ágiles

**Elaborado por:** Fabiana Jaramillo

	<b>Metodologías</b>		
	<b>SCRUM</b>	<b>KANBAN</b>	<b>XP</b>
<b>Características</b>	<ul style="list-style-type: none"> <li>- Para proyectos pequeños, medianos, y grandes</li> <li>- Iteraciones de 1 semana a 1 mes</li> </ul>	<ul style="list-style-type: none"> <li>- Para proyectos medianos y pequeños</li> <li>- Iteraciones continuas</li> </ul>	<ul style="list-style-type: none"> <li>- Para proyectos medianos y pequeños</li> <li>- Iteraciones de 1 semana a 3 semanas</li> </ul>

<b>Fases o Marco de trabajo</b>	<ul style="list-style-type: none"> <li>- Planificación del Sprint,</li> <li>- Etapa de desarrollo,</li> <li>- Revisión del Sprint,</li> <li>- Retroalimentación</li> </ul>	<ul style="list-style-type: none"> <li>Por hacer</li> <li>- Lista de tareas</li> <li>En proceso</li> <li>- Desarrollo</li> <li>- Pruebas</li> <li>- Despliegue</li> <li>Hecho</li> <li>- Finalizado</li> </ul>	<ul style="list-style-type: none"> <li>- Planificación del proyecto</li> <li>- Diseño</li> <li>- Codificación</li> <li>- Pruebas</li> </ul>
<b>Documentación</b>	<ul style="list-style-type: none"> <li>- Product Backlog</li> <li>- Sprint Backlog</li> <li>- Gráfica burndown</li> <li>- Historia de usuarios</li> <li>- Definiciones de hecho</li> </ul>	<ul style="list-style-type: none"> <li>- Tablero o tarjetas</li> <li>Kanban</li> </ul>	<ul style="list-style-type: none"> <li>- Historias de usuario</li> <li>- Tarjetas CRC</li> <li>- Pruebas unitarias de integración y aceptación</li> </ul>

De acuerdo con el análisis comparativo de la **Tabla 9**, se seleccionó la metodología Kanban, debido a que posee procesos bien definidos a cumplirse periódicamente, permite actualizar las tareas fácilmente y además hace un seguimiento de las listas de tareas por hacer, en proceso, y finalizadas.

Para aplicar esta metodología se utilizó el gestor de proyectos Trello. Trello logra una correcta elaboración de desarrollo del proyecto. En Trello se generan tarjetas de actividades, donde se podrá llevar un control y seguimiento de las tareas pendientes, en proceso y hechas.



**Figura 17.** Entorno de trabajo de Trello

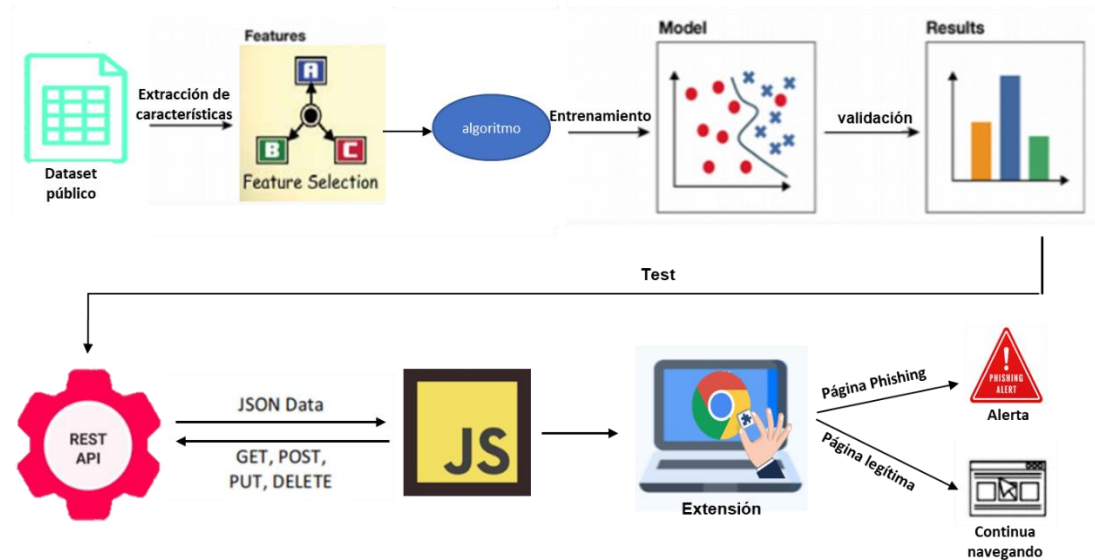
**Elaborado por:** Fabiana Jaramillo

### **3.2.3. Aplicación de la metodología de desarrollo**

#### **3.2.3.1. Proceso de desarrollo de la propuesta**

Para empezar, se procede a recolectar datos mediante dataset públicos. Seguido de la extracción de características obtenidas de la URL. Para luego, continuar con el preprocesamiento de los datos, mejor conocido como limpieza de datos. Los pasos a seguir de esta preparación de datos son: eliminar datos perdidos, categorizar los valores de las variables y normalizar los valores numéricos. Una vez preprocesado los datos se elige el algoritmo más adecuado para la detección de páginas web de Phishing. Elegido el algoritmo se separan los datos preprocesados en datos para entrenamiento y datos para testeo. El modelo de Machine Learning será supervisado, ya que el conjunto de datos para entrenamiento que se generará estará etiquetado. Por lo tanto, el modelo buscará predecir la etiqueta de datos nuevos y no vistos anteriormente. A continuación, se evalúan y validan los resultados obtenidos. Y por último, se lleva a

cabo el proceso para generar una extensión en el navegador con la implementación del modelo de Machine Learning de detección de Phishing ya validado.



**Figura 18.** Esquema del proceso de desarrollo de la propuesta

**Elaborado por:** Fabiana Jaramillo

### 3.2.3.2. Adquisición de datos

Para la adquisición de datos se ha investigado varios conjuntos de datos que contengan URLs de páginas web legítimas como URLs de páginas web Phishing. El primer conjunto de datos utilizado es majestic, compuesto por 1 millón de URLs legítimas [26]. Las URLs Phishing fueron extraídas del conjunto de datos openphish con 6307 URLs Phishing [27], el conjunto de datos ebuDataPhishing con 37176 [28] y phish\_core con 49266 [29]. En los conjuntos de datos de URLs Phishing existe algunas páginas que han dejado de funcionar, ya que han sido detectadas y categorizadas. El conjunto de datos final es construido con un análisis y preparación previa de los datos como: la eliminación de URLs repetidas, para proceder a realizar la extracción de características de cada una. La **Tabla 10.** muestra los conjuntos de datos utilizados.

**Tabla 10.** Conjunto de datos utilizados

**Elaborado por:** Fabiana Jaramillo

	<b>Fuente</b>	<b>N° URLs</b>	<b>N° URLs Total</b>	<b>Nombre del archivo</b>
<b>Legítimas</b>	majestic	1000000	30000	.csv majestic_million
<b>Phishing</b>	openphish	6307	614	.csv openphish
	ebuDataPhishing	37176	10186	.csv ebuDataPhishing
	phish_core	49266	19200	.csv phish_score

### 3.2.3.3. Extracción de Características

Hay que aclarar que no hay una cantidad específica de características que sea recomendable extraer de una URL. Depende de la finalidad de la extracción de características y del contexto en el que se utilizará la URL. Por ejemplo, para este estudio la URL se va a utilizar para clasificar páginas web legítimas y Phishing, por lo que es posible extraer características como la presencia de palabras clave específicas, la longitud de la URL, el número de enlaces externos, la presencia de caracteres sospechosos o la presencia de una cadena de búsqueda malintencionada, etc.

Para sacar las características de una URL primero hay que conocer su estructura, que se muestra en la **Figura 19**.



**Figura 19.** Estructura de la URL

**Elaborado por:** Fabiana Jaramillo

La estructura servirá de guía para poder dividir la URL. A cada una de las URLs se le tratará como una cadena de caracteres que permitirá extraer una gran cantidad de características relevantes para la detección de actividades ilícitas. Los conjuntos de



datos recolectados se representan en modo de características para poder entrenar el modelo de Machine Learning.

Al principio, se extrajeron un total de 38 características de las páginas web recopiladas, 24 de las cuales se obtuvieron de la URL, 13 del contenido de la página web y 1 que es la etiqueta que indica si es una URL Phishing o legítima. Al final se optó por trabajar solo con las 24 características obtenidas de la estructura de la URL y descartar las 13 características que extraían el contenido de la página web, por varios inconvenientes encontrados como:

Primero, muchas páginas web utilizan medidas de seguridad para evitar que los bots accedan a ellas, lo cual causaba problemas al momento de intentar extraer información automáticamente. Segundo, si el código HTML de una página no se encontraba bien estructurado, dificultaba la correcta lectura del mismo, causando problemas al momento de analizarlo con herramientas como BeautifulSoup o lxml de Python. Además, el uso de funciones de extracción de contenido también puede violar los términos de servicio de un sitio web o incluso infringir derechos de autor, por lo que es importante tener en cuenta estas consideraciones legales al utilizar estas funciones.

Las características extraídas para este estudio se muestran en la **Tabla 11**.

**Tabla 11.** Características de la URL

**Elaborado por:** Fabiana Jaramillo

<b>Extracción de características</b>		
<b>URL</b>	Dirección IP	Seguridad SSL (Seguridad de la capa de transporte)
	Número de puntos (.)	Longitud del host
	Longitud de la consulta	Longitud de la ruta del archivo
	Longitud total de la URL	Entropía alfabética
	Tasa de continuidad de caracteres	Número de guiones (-) en el host

	Número de caracteres especiales	Letra-dígito-letra y dígito-letra-dígito
	Número de @ en la URL	Número de letras en el host
	Número de dígitos en el host	Número de guiones bajos ( _ )
	Presencia de www	Número de palabras sospechosas <i>“include signin login ebay account secure confirm bank logon cmd admin paypal”</i>
	Recuento de Top Level Domain (TLD)	Host o IP codificado – forma hexadecimal o base64
	Ofuscación usando forma hexadecimal – en la ruta “%”	Código unicode en la URL
	Archivo ejecutable o no	Ofuscación para redirección
<b>Etiqueta</b>	Si la página es Phishing o no	

Para la extracción de las características se utilizó la librería `urllib` de Python. La librería analiza la estructura de la URL y obtiene la parte a considerar en cada función creada. Además, se hizo uso de expresiones regulares, también conocidas como `regex` o `regexp`. Las expresiones regulares son un lenguaje de patrones utilizado para buscar y reemplazar texto. En este estudio, sirven para buscar patrones específicos en la URL, y verifica si la URL dada cumple con un determinado patrón. (Ver **Anexo C**)

### **Dirección IP**

Los phishers utilizan direcciones IP en sus páginas Phishing, para evitar pagar por un nombre de dominio. [30]

La primera característica se la representa determinando si una URL tiene una dirección IP en su estructura. Se usa una expresión regular (`regex`) para buscar un patrón de dirección IP en la cadena de la URL. Si se encuentra una coincidencia la función retorna 2, caso contrario retorna 1.

## **Seguridad SSL**

Al igual que en la primera característica, los phishers utilizan sitios web sin seguridad para evitar costos extras. La seguridad SSL es una característica relevante para identificar una página web Phishing. [30]

La característica se la representa determinando si una URL tiene seguridad SSL en su estructura y obteniendo información de ella, como el esquema o protocolo (http o https). Si el esquema de la URL es “https” la función retorna 2, de lo contrario retorna 1.

## **Número de puntos (.)**

Otra característica importante que se puede utilizar para identificar páginas web fraudulentas es el número de puntos en la URL de la página. Algunos sitios de phishing suelen utilizar subdominios populares de sitios legítimos para dar la impresión de ser confiables. Esto resulta en un aumento en el número de puntos en la URL de la página. Por lo tanto, las URL con un gran número de puntos tienen más posibilidades de ser sitios fraudulentos. [30]

La función para extraer esta característica realiza un recuento de los puntos encontrados en la URL.

## **Longitud de la URL**

Los sitios de Phishing tienden a tener URLs más extensas. Los creadores de sitios Phishing a menudo tratan de ocultar sus URLs sospechosas utilizando subdominios que parecen legítimas con URLs muy largas. [30]

Para extraer esta característica la estructura de la URL se la divide en tres secciones: el host, la ruta del archivo y la consulta, cada una de ellas se las representa con su propia característica y su longitud específica, como se detalla a continuación: **longitud del host, longitud de la ruta del archivo, longitud de la consulta** y por último la **longitud total de la URL**.

Las funciones de cada una de las características utilizan el módulo `urllib` de Python. Este módulo permite analizar la estructura de una URL y obtener información sobre ella, como el host, la ruta del archivo y la consulta. Si la longitud en la ruta y la consulta son vacías, las funciones devolverán 0.001 caso contrario, devolverán las longitudes correspondientes.

### **Entropía alfabética**

A menudo, los atacantes modifican los caracteres en el dominio con el fin de confundir al usuario y hacer que el dominio parezca legítimo. Por ejemplo, cuando se reemplaza una letra "o" con el número "0" en un dominio como "facebook" y se convierte en "facebo0k". Estos cambios bruscos en los caracteres generan una alteración en la entropía alfabética. [22]

La entropía alfabética se calcula contando el número de ocurrencias de cada carácter en la cadena y utilizando la siguiente fórmula de la entropía.

$$H = - \sum_{i=0}^{25} P(x_i) \log P(x_i)$$

Donde ( $X_0=a, X_1=b, \dots, X_{25}=z$ )

La función que extrae esta característica utiliza el módulo `urllib` de Python para analizar la estructura de la URL, obtener el host y aplicar la fórmula de la entropía para calcular la entropía alfabética del host.

### **Tasa de continuidad de caracteres**

Los propietarios de sitios web legítimos suelen utilizar dominios sencillos y fáciles de recordar. Sin embargo, los atacantes no tienen en cuenta esta característica y crean dominios aleatorios que no les cuesta mucho dinero. La tasa de continuidad de caracteres se utiliza para calcular la longitud más larga de cada tipo de carácter en el nombre del dominio. Esto permite determinar la secuencia de caracteres, dígitos y símbolos en el nombre del dominio. [22]

La característica se la representa contando el número de caracteres consecutivos y dividiéndolo por la longitud total de la cadena. Así mismo la función utiliza el módulo `urllib` de Python para analizar la estructura de la URL y obtener la parte a considerar para el cálculo de la tasa de continuidad.

### **Número de guiones (-) en el host**

El número de guiones en el host de una URL es importante para identificar si un sitio web es de Phishing o no. Los sitios de Phishing a menudo utilizan nombres de host con varios guiones para enmascarar el nombre real del sitio, haciendo que sea difícil para los usuarios detectar el verdadero nombre del dominio.

La función para extraer esta característica hace uso del módulo `urllib` de Python y cuenta el número de guiones en el host usando la función `count` de Python.

### **Número de caracteres especiales**

El número de caracteres especiales en una URL puede ser importante para identificar si un sitio web es de Phishing o no. Los sitios de Phishing a menudo utilizan caracteres especiales para enmascarar el nombre del dominio o el objetivo del sitio. Además, los sitios de phishing también utilizan caracteres especiales para generar nombres de dominio y subdominios que se parezcan a los nombres legítimos, generando confusión en el usuario.

La función que se ha propuesto cuenta el número de caracteres especiales que hay en el host de una URL usando una expresión regular. La expresión regular que se usa es `r'^[\w]'`, que significa "cualquier carácter que no sea una letra, número o guion bajo".

### **Letra-dígito-letra y dígito-letra-dígito**

Para identificar URLs maliciosas que se asemejan a URLs legítimas, se utilizan patrones específicos que implican el uso de números para imitar URLs de sitios web legítimos. Los patrones utilizados son: letra-dígito-letra y dígito-letra-dígito. Esto ayuda a detectar URLs maliciosas que son similares a las URLs benignas. [22]

Para extraer de la estructura de la URL el patrón "letra-dígito-letra" o "dígito-letra-dígito", se usa una expresión regular para buscar estos patrones en el host de la URL.

### **Número de @ en la URL**

Los sitios de Phishing a menudo utilizan @ en su URL para engañar a los usuarios y hacerles creer que están ingresando a un sitio legítimo, mientras que los sitios legítimos raramente utilizan @ en su URL. Para extraer esta característica se usa una expresión regular para buscar este símbolo en la URL.

### **Número de letras en el host**

Los sitios de phishing a menudo utilizan nombres de host con un número reducido de letras para ocultar el nombre real del sitio, mientras que los sitios legítimos tienden a tener nombres de host con un número más alto de letras.

Para extraer esta característica se usa la función len de Python y una expresión regular para contar el número de caracteres alfabéticos encontrados en el host. Finalmente, la función devuelve el número de letras como resultado.

### **Número de dígitos en el host**

Los sitios web Phishing a menudo utilizan nombres de host con un número alto de dígitos para ocultar el nombre real del sitio y engañar a los usuarios. Por otro lado, los sitios legítimos tienden a tener nombres de host con un número bajo de dígitos, con el objetivo de ser fáciles de recordar y reconocer. El uso excesivo de dígitos en la URL puede ser un indicador de que el sitio web es Phishing.

Para extraer esta característica, se usa una expresión regular para buscar dígitos en el host. La función cuenta el número de dígitos encontrados y finalmente, devuelve el número de dígitos como resultado.

### **Número de guiones bajos ( \_ )**

Los sitios de Phishing a menudo utilizan guiones bajos para ocultar el nombre real del sitio o generar subdominios y subdirectorios que se asemejan a los nombres legítimos, mientras que los sitios legítimos raramente utilizan guiones bajos en sus URLs.

Para extraer esta característica se usa una expresión regular para buscar guiones bajos en el host. La función cuenta el número de guiones bajos encontrados y lo devuelve como resultado.

### **Presencia de www**

Los sitios web legítimos utilizan “www” como prefijo en su URL, mientras que los sitios de Phishing a menudo evitan utilizarlo, ya que es fácilmente reconocible y puede ayudar a los usuarios a identificar que el sitio no es legítimo.

Para extraer esta característica se usa la librería `urllib` para analizar la estructura de la URL y obtener el host. Luego, se utiliza el método `startswith` para comprobar si el host comienza con “www”. Finalmente, la función devolverá 2 si comprueba que el host comienza con “www” o 1 en caso contrario.

### **Número de palabras sospechosas**

Los sitios de phishing a menudo utilizan palabras sospechosas en sus URLs, tales como "banco", "PayPal", "cuenta", "iniciarsesión" entre otras, con el objetivo de engañar a los usuarios para ingresar información personal o financiera en sitios web falsos [22]. Esas palabras son utilizadas para generar un efecto visual en el usuario, generando confianza y dando la sensación de que están ingresando a un sitio web legítimo.

Para realizar la función de extracción de esta característica, se define de forma manual un conjunto de palabras clasificadas como sospechosas. Este conjunto de palabras hace referencia a ***“include/signin/login/ebay/account/secure/confirm/bank/logon/cmd/admin/paypal”***. Luego se utiliza los métodos `count` para contar el número de veces que cada palabra sospechosa aparece en el host y en la ruta del archivo. Finalmente, la función devuelve el número total de palabras sospechosas encontradas.

### **Recuento de Top Level Domain (TLD)**

El recuento de TLD en el host de una URL puede ser importante para identificar si un sitio web es de phishing o no. Las URLs Phishing suelen tener múltiples TLDs generalmente no reconocidos o no comunes para ocultar el nombre real del sitio.

Para extraer esta característica la función devuelve el recuento de TLD de último nivel que cuenta el punto desde el final del host en la URL.

### **Host o IP codificado – forma hexadecimal o base64**

El uso de host o IP codificado en forma hexadecimal o base64 en una URL puede ser un indicador de que un sitio web es de Phishing. Los sitios de phishing a menudo utilizan codificación para ocultar el nombre real del sitio o para generar subdominios y subdirectorios que se asemejan a los nombres legítimos. El uso de codificación en lugar de un nombre de dominio legible puede ser un indicador de que el sitio web no es legítimo y está tratando de ocultar su verdadera intención. [22]

Para extraer esta característica se usa una expresión regular en la función que comprueba si el host está compuesto únicamente por dígitos hexadecimales (0-9 y a-f) o por caracteres válidos para la codificación base64 (A-Z, a-z, 0-9, +, / y =). Si el host cumple alguno de estos criterios, la expresión regular encuentra una coincidencia y la función devuelve 2. Si no encuentra ninguna coincidencia, la función devuelve 1.

### **Ofuscación usando forma hexadecimal – en la ruta “%”**

La ofuscación en la ruta de una URL utilizando forma hexadecimal, como el carácter “%”, puede ser un indicador de que un sitio web es Phishing. Los sitios de phishing a menudo utilizan ofuscación para ocultar el nombre real de la ruta del sitio o en su defecto introducir algún malware.

Para extraer esta característica se usa una expresión regular que comprueba si en la ruta de la URL se encuentra al menos un carácter codificado en forma hexadecimal. Si la función encuentra aquella codificación devuelve 2, caso contrario devuelve 1.



### **Código unicode en la URL**

El uso de caracteres unicode en una URL puede ser un indicador de que un sitio web es Phishing. Los sitios de phishing a menudo utilizan caracteres unicode para crear subdominios y subdirectorios que se asemejan a los nombres legítimos, pero con caracteres unicode que se ven similares a los caracteres normales. Esto puede engañar a los usuarios para que ingresen información personal o financiera en sitios web falsos.

Para extraer esta característica se usa una expresión regular que busca cualquier secuencia de caracteres que comiencen con %u. Si la función encuentra una coincidencia devuelve 2 caso contrario 1.

### **Archivo ejecutable o no**

Los sitios de Phishing a menudo incluyen enlaces a archivos ejecutables que, una vez descargados e instalados en el dispositivo del usuario, pueden instalar malware o robar información. Los sitios legítimos, por otro lado, no suelen incluir enlaces a archivos ejecutables. [22]

Para extraer esta característica se usa la función `urlparse` del módulo `urllib` de Python para analizar la estructura de la URL y obtener el nombre del archivo. Luego, se compara el nombre del archivo con una lista de extensiones de archivos ejecutables comunes, como ".exe", ".com", o ".bat". Si la función encuentra una coincidencia devuelve 2 caso contrario 1.

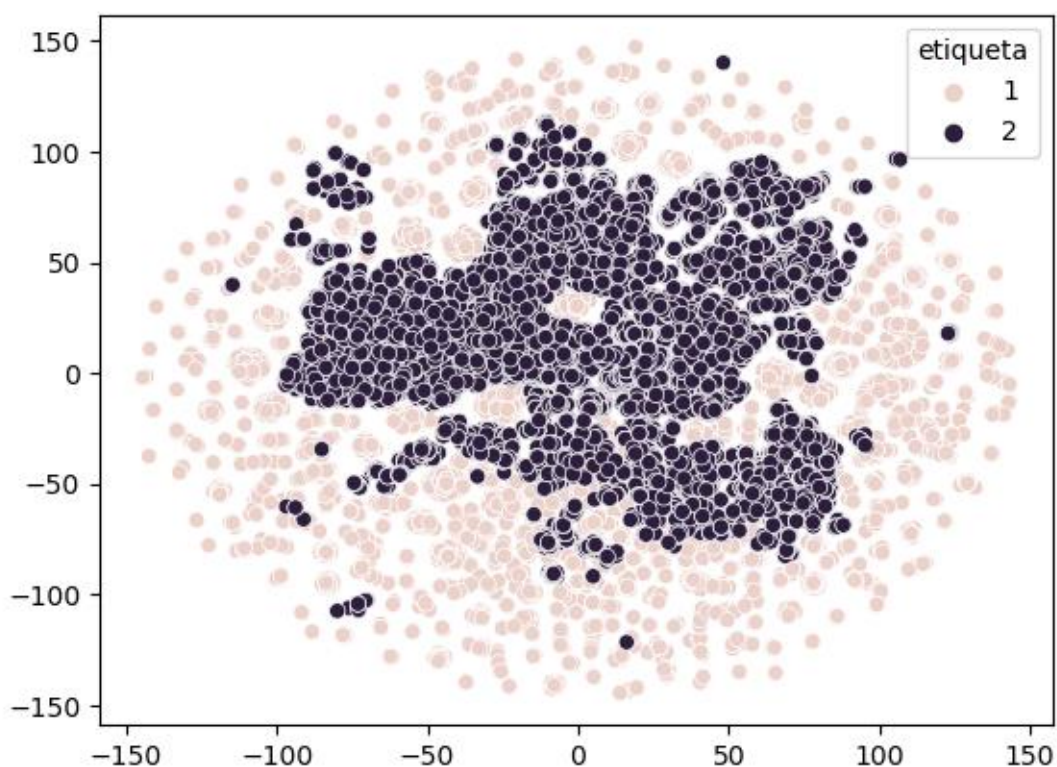
### **Ofuscación para redirección**

La ofuscación de redirección es una técnica que se utiliza para hacer que una URL aparentemente legítima redirija a una página maliciosa o a otra URL no deseada. Un ejemplo común de ofuscación de redirección es el uso de caracteres especiales como el @ para ocultar la verdadera dirección de la página de destino.

Por lo tanto, la función que extrae esta característica usa una expresión regular para buscar cualquier secuencia de caracteres que comiencen con @. Si la función encuentra una coincidencia devuelve 2 caso contrario 1.

### 3.2.3.4. Representación del dataset final

Una vez extraídas las 24 características de todas las URLs adquiridas de los distintos dataset, se forma el dataset final para empezar con el entrenamiento del modelo. Para explorar y entender mejor los patrones y las características del dataset final se graficó y usó el algoritmo t-SNE (T-distributed Stochastic Neighbor Embedding). El objetivo del algoritmo es maximizar la similitud entre los puntos cercanos en el espacio de alta dimensión y el espacio de baja dimensión. El gráfico t-SNE ayuda a entender y visualizar mejor las relaciones entre las características del dataset, siendo útil para identificar patrones y estructuras en los datos. El gráfico t-SNE se muestra en la **Figura 20**.



**Figura 20.** Gráfico t-SNE

**Elaborado por:** Fabiana Jaramillo

El gráfico t-SNE muestra los puntos de la etiqueta 2, que representan URLs Phishing, agrupados en el centro y los puntos de la etiqueta 1, que representan URLs legítimas, rodeándolos y distribuidos en una zona más amplia. La distribución indica que las URLs Phishing tienen una mayor similitud entre sí en términos de características, y que las URLs legítimas tienen una mayor variedad de características. Esto puede ser

útil para identificar patrones o características comunes en las URLs Phishing que podrían ser utilizadas para detectar futuras URLs Phishing.

### **3.2.3.5. Elección del algoritmo**

Los algoritmos de Machine Learning que se analizaron para realizar el modelo de detección de Phishing fueron: Random Forest (RF), Support Vector Machine (SVM), Red Neuronal Artificial (RNA) y K-Nearest Neighbors (KNN).

- **Random Forest (RF)**

Random Forest es un algoritmo de clasificación y regresión que utiliza una colección de árboles de decisión para realizar predicciones [22]. La estructura básica del algoritmo es la siguiente: Primero, la creación de un número específico de árboles de decisión utilizando un subconjunto aleatorio de los datos de entrenamiento. Segundo, cada árbol se construye utilizando un subconjunto aleatorio de las características de los datos de entrenamiento. Tercero, cada árbol hace una predicción independiente para cada entrada. Y, por último, las predicciones de todos los árboles se combinan para determinar la predicción final, generalmente mediante una votación entre las predicciones de cada árbol [31]. La estructura de Random Forest ayuda a prevenir el sobreajuste y mejorar la capacidad generalizadora del modelo. [30]

- **Support Vector Machine (SVM)**

Support Vector Machine es un algoritmo de aprendizaje automático supervisado que se utiliza para clasificación y regresión. Se basa en la idea de encontrar un hiperplano óptimo en un espacio de características de alta dimensión que separa de manera óptima diferentes clases de datos. En términos estructurales, SVM consta de un conjunto de vectores de soporte, que son los puntos de datos más cercanos al hiperplano, y un hiperplano de separación que maximiza la distancia entre los vectores de soporte de las diferentes clases. [22]

- **Red Neuronal Artificial (RNA)**

Las Redes Neuronales Artificiales son una clase de algoritmos de aprendizaje automático inspirados en la estructura y funcionamiento del cerebro humano. Se estructura como un montón de neuronas interconectadas, que procesan y transmiten información. Se compone de tres capas: la capa de entrada, la capa oculta y la capa de salida. La capa de entrada recibe los datos de entrada y los procesa a través de una serie de pesos y umbrales, que determinan qué información se envía a la capa oculta. La capa oculta procesa la información recibida y la transmite a la capa de salida, donde se genera una respuesta o salida.[32]

- **K-Nearest Neighbors (KNN)**

El algoritmo k-Nearest Neighbors (KNN) es un método de clasificación basado en la idea de que un punto dado es similar a los puntos que lo rodean en el espacio de características. Es un algoritmo de aprendizaje no paramétrico, es decir, no se asume ninguna distribución específica para los datos. El algoritmo es considerado un método “perezoso” por utilizar los datos de entrenamiento para generar un modelo generalizado y utilizarlo para clasificar. Cuando se presenta un nuevo ejemplo observado, se calcula la distancia entre ese ejemplo y todos los ejemplos de entrenamiento almacenados, y se seleccionan los k ejemplos más cercanos. El valor k es un número de vecinos más cercanos con características similares para realizar la comparación y llevar a cabo la elección. La clase a la que pertenece el nuevo ejemplo se determina a partir de la mayoría de las clases de los k vecinos más cercanos. [33]

A continuación, se muestra la **Tabla 12**, que es una comparativa de los algoritmos previamente entrenados, con la división del dataset original, su estructura y métricas correspondientes.

**Tabla 12.** Tabla comparativa de los algoritmos

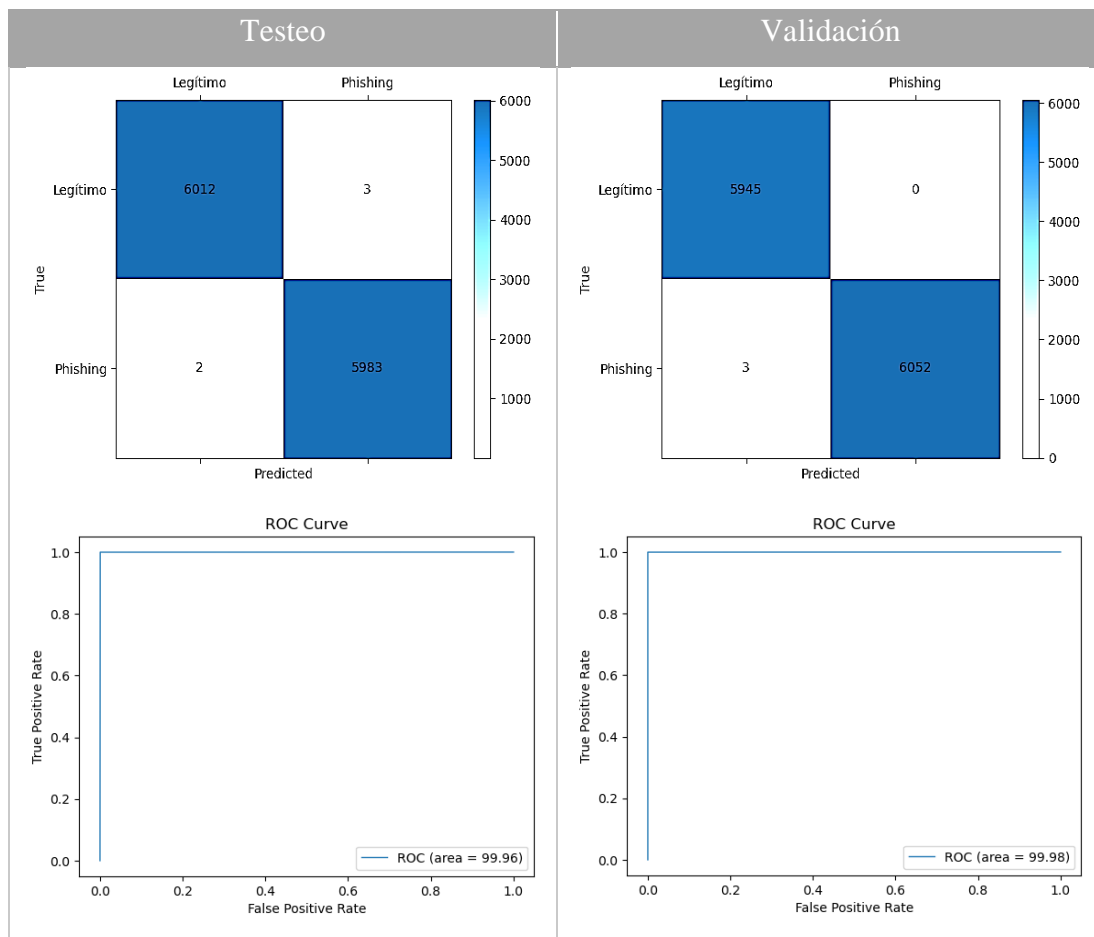
**Elaborado por:** Fabiana Jaramillo

<b>Algoritmo</b>	<b>División del dataset</b>	<b>Estructura</b>	<b>Métricas</b>
<b>Random Forest (RF)</b>	Se divide en 70% de entrenamiento, 15% de prueba y 15% de validación.	El clasificador se crea con 100 árboles de decisión, y una profundidad máxima de 7 en cada árbol.	Entrenamiento precisión = 99.97% Testeo precisión = 99.96% exactitud = 99.96% ROC = 99.96% Validación precisión = 99.96% exactitud = 99.98% ROC = 99.98%
<b>Support Vector Machine (SVM)</b>	Se divide en 70% de entrenamiento, 15% de prueba y 15% de validación.	Se crea un objeto de la clase SVC con el kernel 'rbf' y un parámetro gamma=7.	Entrenamiento precisión = 100% Testeo precisión = 98.07% exactitud = 99.04% ROC = 99.03% Validación precisión = 97.98% exactitud = 98.99% ROC = 98.99%
<b>Red Neuronal Artificial (RNA)</b>	Se divide en 60% de entrenamiento, 20% de prueba y 20% de validación.	Red neuronal artificial de tipo secuencial de 4 capas densas. La primera de 25 unidades recibe una entrada de 24 dimensiones, utilizando una función de activación ReLU. La segunda y tercera capas	Entrenamiento exactitud = 99.92% Testeo precisión = 99.95% exactitud = 99.96% ROC = 99.96% Validación precisión = 100% exactitud = 99.98%

		<p>tienen 15 y 7 unidades respectivamente, también utilizando una función de activación ReLU. La última capa tiene 3 unidades y utiliza una función de activación softmax para producir una salida de probabilidades.</p>	<p>ROC = 99.98%</p>
<p><b>K-Nearest Neighbors (KNN)</b></p>	<p>Se divide en 70% de entrenamiento, 15% de prueba y 15% de validación.</p>	<p>Se crea un objeto de la clase KNeighborsClassifier, especificando el número de vecinos más cercanos a considerar en la clasificación como 1.</p>	<p>Entrenamiento  precisión = 100%</p> <p>Testeo  precisión = 99.64%  exactitud = 99.82%</p> <p>ROC = 99.82%</p> <p>Validación  precisión = 99.69%  exactitud = 99.78%  ROC = 99.78%</p>

De acuerdo con el análisis comparativo de la **Tabla 11**. Se seleccionó el algoritmo de la Red Neuronal Artificial (RNA), debido a que tiene una alta precisión del 100% y exactitud del 99.98%, lo que significa que el modelo es capaz de clasificar correctamente la mayoría de las muestras en la validación, y no solo eso, sino que también tiene un ROC (Receiver Operating Characteristic) del 99.98% lo que significa que la capacidad de separar las dos clases es muy alta. Además de esto, la RNA es un tipo de modelo altamente flexible y poderoso, capaz de manejar grandes conjuntos de datos y características complejas, lo que permite una mejor capacidad de adaptarse a patrones y relaciones en los datos. En este caso, se especifica que tiene una arquitectura de 4 capas densas, lo que permite una buena capacidad de representación de los datos.

La **Figura 21.** muestra la matriz de confusión y el ROC del modelo RNA de los datos de testeo y validación.



**Figura 21.** Matriz de confusión y ROC del modelo RNA de los datos de testeo y validación

**Elaborado por:** Fabiana Jaramillo

### 3.2.3.6. Desarrollo de la extensión

Para el desarrollo de la extensión al navegador se procede a realizar lo siguiente:

- **Guardar el modelo entrenado**

Para guardar el modelo entrenado se ha utilizado la librería `joblib` de Python. La librería `joblib` almacena y carga objetos en Python de manera eficiente para volver a ejecutarlo si es necesario. También es compatible con la mayoría de las funciones de Python y soporta compresión y paralelismo. Además, `joblib` es compatible con la

mayoría de las bibliotecas de aprendizaje automático populares como scikit-learn, TensorFlow, etc. [34]

En este caso, la librería joblib guarda el objeto del Moledo de entrenamiento RNA usando la función dumb. La función dumb guarda el objeto en un archivo llamado modeloRN.joblib. Esto permite usar el modelo RNA entrenado sin tener que volver a entrenarlo. (**Anexo D**)

- **Crear la función predecir**

La función predecir se crea para realizar las predicciones sobre una URL específica. La función carga el modelo RNA previamente entrenado y guardado con el nombre modeloRN.joblib utilizando la función load de la librería joblib. Seguidamente, la función llama a otra función de nombre url\_has\_vector que retorna el vector de características de la URL. Para terminar, se utiliza el modelo para hacer una predicción sobre el vector de características de la URL dada, utilizando el método predict que retorna la etiqueta predicha. (**Anexo E**)

- **Crear una API RESTful utilizando Flask**

La API RESTful se crea utilizando Flask, un marco web de Python, que acepta solicitudes y respuestas HTTP. Para empezar, se realiza una función controladora que maneja las solicitudes en la ruta "/rna". Dentro de la función se extrae el parámetro "url" de la solicitud GET, y se utiliza para llamar a la función "predecir(url)" que devuelve la respuesta en formato JSON. (**Anexo F**)

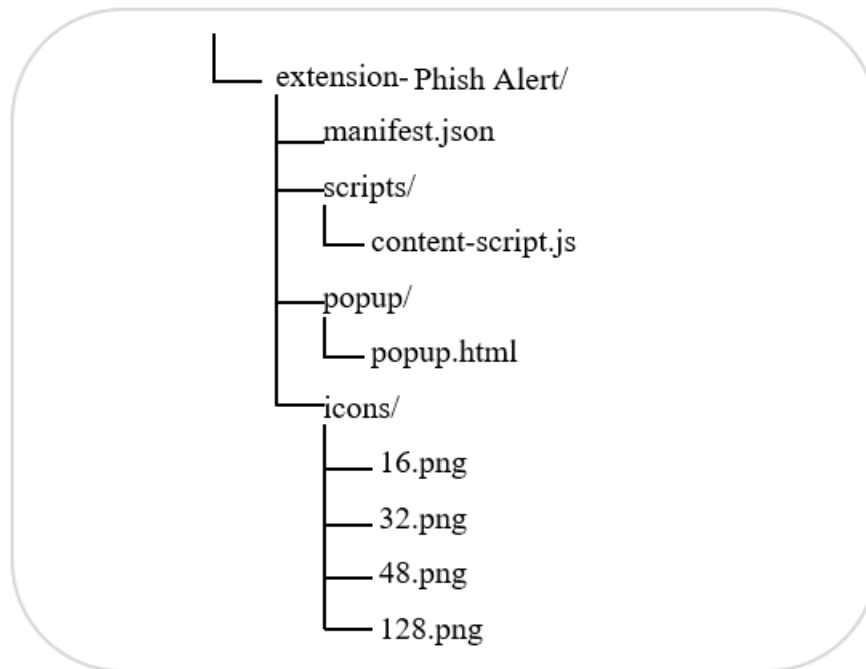
- **Crear la arquitectura de la extensión**

Se crea una extensión de Chrome que se compone de los siguientes elementos:

- Un archivo manifest.json que contiene información básica sobre la extensión, como su nombre, descripción y versión.
- Un archivo HTML y un archivo JavaScript que proporciona la interfaz de usuario y funcionalidad de la extensión.
- Un recurso tipo iconos.



La **Figura 21.** muestra la arquitectura de la extensión, nombrada Phish Alert.



**Figura 22.** Estructura de la extensión Phish Alert

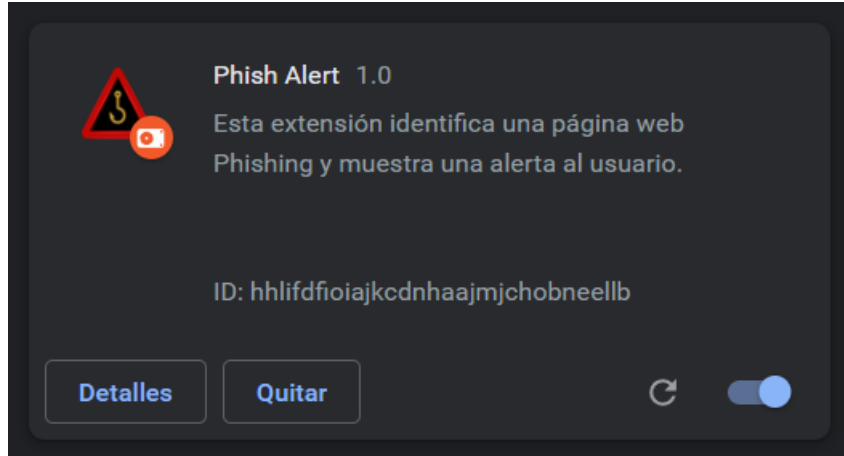
**Elaborado por:** Fabiana Jaramillo

### 3.2.3.7. Funcionalidad de la extensión

El archivo de JavaScript que proporciona la funcionalidad de la extensión utiliza la función `fetch` para enviar una solicitud GET al servidor local “http://localhost:5000/rna” con un parámetro de consulta “url” que se establece en la URL actual de la página web. Se espera que el servidor responda con un objeto JSON que contenga un campo llamado “data”. Si el valor de este campo es “Legítima”, se registra un mensaje en la consola que dice “Funciona, es una página legítima” de lo contrario, se muestra una alerta al usuario diciendo “Esta página parece ser Phishing te recomiendo salir”. Si hay un error con la solicitud, registrará el mensaje de error en la consola. (**Anexo G**)

En resumen, esta extensión es una alerta de phishing que identifica sitios web fraudulentos y muestra una alerta al usuario. Se utilizó un archivo de manifest para especificar la acción predeterminada, las imágenes del icono y los scripts de contenido que se ejecutan automáticamente en todas las páginas web visitadas. (**Anexo H**)

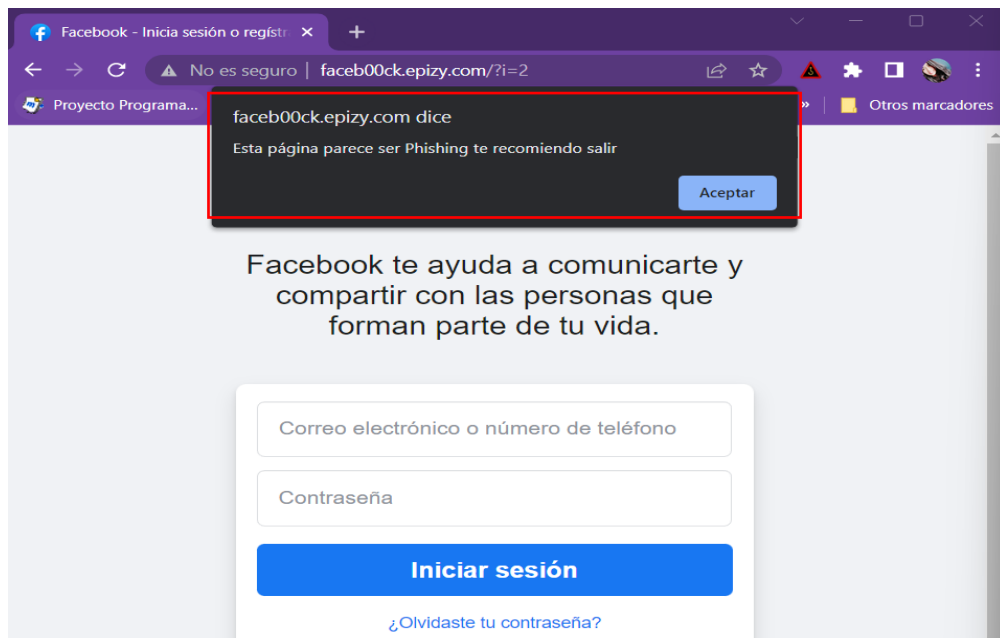
La **Figura 22.** Muestra la extensión subida y activada en el navegador.



**Figura 23.** Extensión subida y activa en el navegador

**Elaborado por:** Fabiana Jaramillo

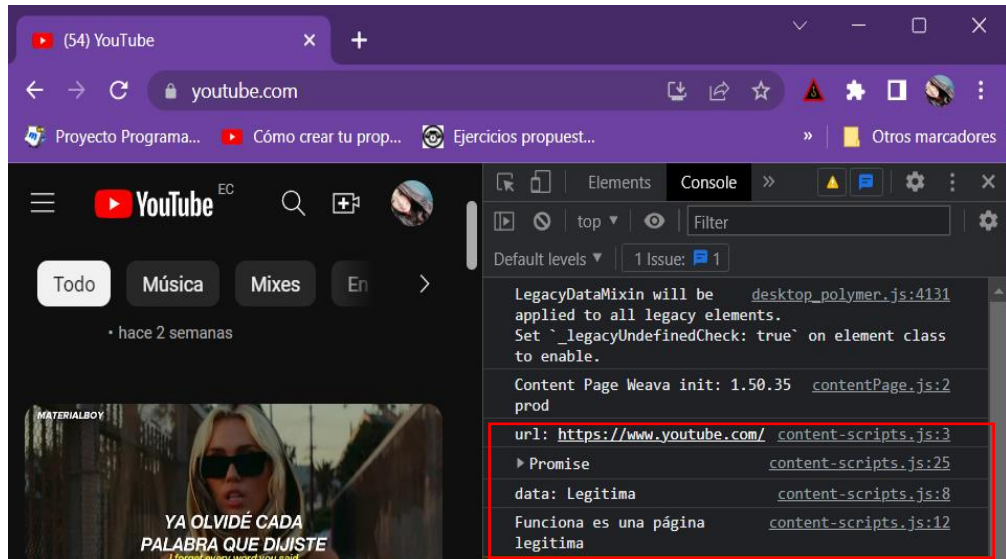
La **Figura 23.** Muestra el funcionamiento de la extensión llamada Phish Alert con una página Phishing.



**Figura 24.** Página Phishing – mensaje de alerta

**Elaborado por:** Fabiana Jaramillo

La **Figura 24**. Muestra el funcionamiento de la extensión llamada Phish Alert con una página Phishing.



**Figura 25.** Página Legítima – mensaje en consola

**Elaborado por:** Fabiana Jaramillo

## **CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones**

- Se determina que la comunidad de la FISEI está altamente propensa a sufrir ataques de Phishing. A pesar de que los usuarios suelen verificar la URL antes de acceder a un enlace, los métodos de ataque actuales hacen que sea difícil para ellos reconocer la veracidad de la URL.
- En cuanto a la clonación de las páginas de facebook y netflix a través de HTTrack se presentaron dificultades debido a la protección contra copias y la utilización de tecnologías avanzadas en las páginas.
- Se determina que el modelo que mejor se ajusta al problema es una Red Neuronal Artificial (RNA) que en los datos de validación tiene una alta precisión, exactitud y ROC lo que indica una buena capacidad de generalización y una buena capacidad de adaptación a los patrones y relaciones en los datos.
- Sobre el desarrollo de la extensión, el mayor grado de complejidad se presenta al momento de hacer solicitudes en este caso con el método GET, por políticas del CORS.

### **4.2. Recomendaciones**

- Se recomienda utilizar una escala de likert en todas las preguntas de la encuesta ya que, ayuda a mantener la consistencia en la recopilación de datos y en la interpretación de los resultados. Las escalas de Likert son fáciles de analizar estadísticamente. Los datos se pueden agrupar y analizar fácilmente para obtener una visión general de las respuestas.
- Para resolver el problema de la clonación de las páginas facebook y netflix es recomendable copiar y pegar manualmente el contenido deseado.

- Se recomienda instalar la biblioteca keras que es específicamente para modelos de redes neuronales. Keras proporciona una interfaz sencilla y fácil de usar para construir, entrenar y evaluar el modelo.
- Se recomienda importar el módulo CORS para manejar las solicitudes cruzadas entre dominios y evitar el bloqueo de las solicitudes. Tomar como base para futuros trabajos el modificar las políticas del CORS desde código y evitar problemas con actualizaciones del navegador.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] J. Julián, M. Martínez, C. Hernán, C. Osorio, and M. Mejía, “Análisis del Phishing y la Ley de delitos informáticos en Colombia,” 2021.
- [2] Nadezhda Demidova, “Los phishers y su sed de conocimiento | Securelist,” Oct. 24, 2018. <https://securelist.lat/phishing-for-knowledge/88014/> (accessed May 18, 2022).
- [3] D. A. H. VERA, “La Suplantación De Identidad Cibernética En El Ecuador,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.
- [4] Luis Fernando Rosero Tejada, “EL PHISHING COMO RIESGO INFORMÁTICO, TÉCNICAS Y PREVENCIÓN EN LOS CANALES ELECTRÓNICOS: UN MAPEO SISTEMÁTICO,” 2021.
- [5] P. F. M. Guangashi, “Medidas De Protección Informática Para Evitar El Robo De Identidad Provocado Por El Ataque Phishing “the Tabnabbing Attack” Para La Facultad De Ingeniería En Sistemas, Electrónica E Industrial,” p. 159, 2012.
- [6] Byron Guerra, “Instituciones educativas en riesgo informático - UDLA,” Dec. 15, 2021. <https://www.udla.edu.ec/liderazgo/blog/2021/12/15/instituciones-educativas-en-riesgo-informatico/> (accessed Sep. 14, 2022).
- [7] A. Ginsberg and C. Yu, “Rapid homoglyph prediction and detection,” *Proc. - 2018 1st Int. Conf. Data Intell. Secur. ICDIS 2018*, pp. 17–23, 2018, doi: 10.1109/ICDIS.2018.00010.
- [8] S. Abdelnabi, K. Krombholz, and M. Fritz, “VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity,” *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 1681–1698, 2020, doi: 10.1145/3372297.3417233.
- [9] H. Suzuki, D. Chiba, Y. Yoneya, T. Mori, and S. Goto, “ShamFinder: An automated framework for detecting IDN homographs,” *Proc. ACM SIGCOMM Internet Meas. Conf. IMC*, pp. 449–462, 2019, doi: 10.1145/3355369.3355587.
- [10] A. M. Almuhaideb *et al.*, “Homoglyph Attack Detection Model Using Machine Learning and Hash Function,” *J. Sens. Actuator Networks*, vol. 11, no. 3, 2022, doi: 10.3390/jsan11030054.
- [11] K. Tian, S. T. K. Jan, H. Hu, D. Yao, and G. Wang, “Needle in a haystack:

- Tracking down elite phishing domains in the wild,” *Proc. ACM SIGCOMM Internet Meas. Conf. IMC*, pp. 429–442, 2018, doi: 10.1145/3278532.3278569.
- [12] P. Deng, C. Linsky, and M. Wright, “Weaponizing Unicodes with Deep Learning -Identifying Homoglyphs with Weakly Labeled Data,” *Proc. - 2020 IEEE Int. Conf. Intell. Secur. Informatics, ISI 2020*, 2020, doi: 10.1109/ISI49825.2020.9280538.
- [13] E. Fidalgo and E. Alegre, “Impact of current phishing strategies in machine learning models for phishing detection,” 2021.
- [14] M. Demartini and M. Rios, “El impacto generado por la seguridad informática en las PYMES de Mendoza.,” pp. 1–49, 2019.
- [15] Jairo Ricardo Zamora Sánchez, “LOS DELITOS INFORMÁTICOS Y EL DERECHO A LA INTIMIDAD EN EL CÓDIGO ORGÁNICO INTEGRAL PENAL,” p. 3060107, 2020.
- [16] Bryan Andrés Guerrón Chiluisa, “POLÍTICAS DE SEGURIDAD INFORMÁTICA PARA EL TELETRABAJO EN LA EMPRESA BIOALIMENTAR,” p. 6, 2021.
- [17] D. A. S. Trujillo, “(1656) Técnicas utilizadas por los ciberdelincuentes y cómo protegerse - YouTube,” 2020. <https://www.youtube.com/watch?v=3sTpkDJBCXY> (accessed May 30, 2022).
- [18] E. Benavides, W. Fuertes, and S. Sanchez, “Caracterización de los ataques de phishing y técnicas para mitigarlos. Ataques: una revisión sistemática de la literatura,” *Cienc. y Tecnol.*, vol. 13, no. 1, pp. 97–104, 2020, doi: 10.18779/cyt.v13i1.357.
- [19] L. Irwin, “Los 5 tipos de ataques de phishing más comunes - IT Governance Blog Es,” 2022. <https://www.itgovernance.eu/blog/en/the-5-most-common-types-of-phishing-attack> (accessed May 29, 2022).
- [20] P. Kalaharsha and B. M. Mehtre, “Detecting Phishing Sites -- An Overview,” pp. 1–13, 2021, [Online]. Available: <http://arxiv.org/abs/2103.12739>
- [21] A. N. del Ecuador, “Código Orgánico Integral Penal (Última Reforma: 16-III-2022),” p. 15, 2022.
- [22] M. Martha and C. Maestre, “Detección de URLs maliciosas por medio de técnicas de aprendizaje,” *Univ. Nac. Colomb.*, 2021.

- [23] E. Manrique, “Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo,” *Rev. Ibérica Sist. e Tecnol. Informação*, pp. 586–599, 2020, [Online]. Available: <https://search.proquest.com/openview/c7e24c997199215aa26a39107dd2fe98/1?pq-origsite=gscholar&cbl=1006393>
- [24] MathWorks, “MATLAB vs. Python ¿Cuál se adapta mejor a sus necesidades? - MATLAB & Simulink.” <https://la.mathworks.com/products/matlab/matlab-vs-python.html> (accessed Jan. 30, 2023).
- [25] Christian Fernando Calo Tisalema, “SISTEMA PARA MINIMIZAR EL RIESGO DURANTE LA ENTREGA DE LOS ESTUDIANTES DE EDUCACIÓN INICIAL A LOS REPRESENTANTES USANDO TÉCNICAS DE RECONOCIMIENTO FACIAL EN LA UNIDAD EDUCATIVA CRISTIANA NEW LIFE,” 2022.
- [26] K. W. Brown, “kyle-w-brown/majestic\_million: Majestic Million dataset,” 2021. [https://github.com/kyle-w-brown/majestic\\_million](https://github.com/kyle-w-brown/majestic_million) (accessed Dec. 21, 2022).
- [27] “OpenPhish - Phishing Intelligence,” 2021. <https://openphish.com/index.html> (accessed Dec. 21, 2022).
- [28] “ebubekirbbr/pdd,” 2020. <https://github.com/ebubekirbbr/pdd> (accessed Dec. 21, 2022).
- [29] “PhishStats,” 2022. <https://phishstats.info/> (accessed Dec. 21, 2022).
- [30] T. O. Ojewumi, G. O. Ogunleye, B. O. Oguntunde, O. Folorunsho, S. G. Fashoto, and N. Ogbu, “Performance evaluation of machine learning tools for detection of phishing attacks on web pages,” *Sci. African*, vol. 16, p. e01165, 2022, doi: 10.1016/j.sciaf.2022.e01165.
- [31] J. Shad and S. Sharma, “A Novel Machine Learning Approach to Detect Phishing Websites Jaypee Institute of Information Technology,” pp. 425–430, 2018.
- [32] M. V. Kunju, E. Dainel, H. C. Anthony, and S. Bhelwa, “Evaluation of phishing techniques based on machine learning,” *2019 Int. Conf. Intell. Comput. Control Syst. ICCS 2019*, no. Iciccs, pp. 963–968, 2019, doi: 10.1109/ICCS45141.2019.9065639.
- [33] J. Zamorano, “Comparativa y análisis de algoritmos de aprendizaje automático



para la predicción del tipo predominante de Cubierta Arbórea,” p. 133, 2018, [Online]. Available: [https://eprints.ucm.es/id/eprint/48800/1/Memoria TFM Machine Learning\\_Juan\\_Zamorano\\_para\\_difundir \(2\).pdf%0Ahttps://eprints.ucm.es/id/eprint/48800/](https://eprints.ucm.es/id/eprint/48800/1/Memoria_TFM_Machine_Learning_Juan_Zamorano_para_difundir(2).pdf%0Ahttps://eprints.ucm.es/id/eprint/48800/)

- [34] “Joblib: running Python functions as pipeline jobs — joblib 1.3.0.dev0 documentation,” 2021. <https://joblib.readthedocs.io/en/latest/> (accessed Jan. 26, 2023).

# ANEXOS

## Anexo A

**Tabla 13.** Alfa de Cronbach

Elaborado por: Fabiana Jaramillo

Encuestados	Ítem 1	Ítem 2	Ítem 3	Ítem 4	Ítem 5	Ítem 6	Ítem 7	Ítem 8	Ítem 9	Ítem 10	Ítem 11	Suma
1	3	2	3	1	1	3	3	3	3	4	3	29
2	3	2	3	1	1	1	3	4	1	5	3	27
3	5	3	4	3	4	4	4	3	3	4	4	41
4	3	4	4	3	3	3	4	4	4	4	5	41
5	2	4	4	3	5	5	5	4	4	5	5	46
6	2	2	4	2	2	2	4	4	1	4	3	30
7	4	4	4	4	4	4	4	4	4	4	4	44
8	3	3	4	4	3	3	4	4	2	4	4	38
9	5	4	5	5	4	5	5	5	4	5	5	52
10	3	2	4	3	3	3	4	4	3	4	3	36
11	2	2	3	5	3	3	3	3	3	3	4	34
12	4	3	4	5	4	3	5	4	4	5	4	45
13	4	3	5	5	4	3	5	5	2	5	5	46
14	3	2	3	1	1	3	2	3	1	4	3	26
15	4	3	5	5	2	4	5	4	4	5	5	46
16	4	3	5	3	2	2	4	5	3	4	3	38
17	3	4	3	4	4	4	3	3	4	3	4	39
18	3	4	4	4	4	3	4	4	3	3	4	40
19	3	4	5	4	5	4	4	3	4	4	5	45
20	3	3	3	1	1	1	5	5	1	4	3	30
21	3	3	5	1	1	3	5	5	3	5	3	37
22	3	2	4	3	2	2	4	5	3	4	3	35
23	3	3	4	3	2	3	4	4	3	4	5	38
24	4	4	4	3	2	4	4	4	4	5	3	41
25	4	4	4	4	5	4	4	4	5	5	5	48
26	3	2	5	1	1	1	5	4	1	5	3	31
27	3	4	3	2	3	4	4	5	4	5	5	42
28	3	1	4	1	1	3	4	4	4	5	3	33
29	3	2	5	3	3	3	4	3	4	4	5	39
30	3	1	3	2	2	2	4	4	3	4	3	31
31	4	2	5	2	4	2	5	4	5	5	5	43
32	3	3	4	4	4	3	3	3	3	3	3	36
33	3	4	3	3	3	3	3	4	4	3	4	37
34	4	2	4	4	2	3	5	4	4	4	3	39
35	3	2	4	1	2	2	4	3	2	5	3	31
36	3	2	4	4	2	4	4	2	3	4	3	35
37	2	2	3	3	2	3	4	4	4	4	4	35
38	3	3	3	1	3	3	4	3	3	5	5	36
39	3	3	5	5	1	3	5	5	5	3	5	43
40	3	2	4	3	1	2	4	3	3	5	5	35
41	3	2	4	3	2	3	5	4	1	4	3	34
42	3	3	3	1	1	3	4	4	1	4	5	32
43	4	3	4	2	1	3	3	4	5	5	5	39
44	4	4	5	3	3	4	4	4	4	5	5	45
45	3	3	4	3	3	3	4	4	3	4	4	38
46	3	4	5	4	3	3	4	4	4	4	5	43
47	1	1	5	1	1	3	5	5	4	5	5	36
48	4	2	4	3	3	3	4	3	3	5	3	37
49	4	2	4	3	5	3	4	5	4	5	5	44

50	2	2	3	1	1	1	3	3	3	4	4	27
51	3	3	4	2	2	2	5	4	3	5	5	38
52	4	3	4	2	1	4	4	4	4	4	5	39
53	3	3	5	3	3	3	4	4	4	4	5	41
54	5	4	4	3	5	3	5	5	3	4	4	45
55	3	4	4	4	4	4	4	4	4	4	5	44
56	3	3	4	3	1	2	4	5	4	4	5	38
57	3	2	4	3	1	2	4	3	3	3	3	31
58	4	3	5	4	3	4	5	3	4	5	5	45
59	3	4	5	5	1	5	4	4	5	5	5	46
60	2	2	4	1	1	4	5	5	4	5	4	37
61	3	3	5	5	1	5	5	3	5	5	5	45
62	2	1	3	1	1	4	5	1	1	3	3	25
63	4	4	5	4	4	4	4	5	3	4	5	46
64	4	4	4	5	5	5	5	4	4	4	5	49
65	4	2	5	1	1	2	4	5	3	5	5	37
66	3	3	4	3	2	4	4	4	4	4	5	40
67	3	3	4	2	2	2	3	3	3	2	4	31
68	3	3	4	2	2	2	3	3	3	4	5	34
69	5	1	4	5	3	4	3	3	4	3	4	39
70	3	2	4	3	2	2	5	3	3	5	3	35
71	2	4	5	3	1	3	5	5	3	5	3	39
72	3	3	5	2	2	5	5	5	5	5	5	45
73	3	2	4	2	3	3	4	4	4	3	3	35
74	3	3	4	4	2	3	3	3	1	1	3	30
75	4	4	3	4	2	3	3	3	3	3	4	36
76	3	2	4	3	4	2	5	5	4	5	5	42
77	3	2	4	3	3	3	4	3	3	3	5	36
78	3	3	3	1	1	3	4	4	2	4	5	33
79	4	3	5	1	1	3	4	4	1	5	5	36
80	4	3	3	3	3	2	4	4	4	4	5	39
81	3	1	4	3	3	3	4	4	4	4	3	36
82	4	2	4	4	2	4	4	4	3	4	5	40
83	3	4	4	4	4	3	5	3	3	4	5	42
84	3	3	4	3	2	4	5	5	2	5	4	40
85	3	2	3	2	2	2	3	3	3	3	3	29
86	4	3	5	2	3	4	5	4	5	5	5	45
87	3	3	4	3	3	4	4	3	3	5	5	40
88	4	3	4	3	4	3	4	4	3	5	5	42
89	3	2	4	2	4	5	5	4	5	4	5	43
90	2	3	4	2	1	2	4	4	3	4	3	32
91	3	3	4	5	3	3	4	4	4	5	5	43
92	5	4	3	5	4	1	5	5	4	3	3	42
93	5	1	3	1	1	3	5	5	4	4	5	37
94	2	1	4	2	1	2	2	4	3	4	3	28
95	4	3	5	5	3	5	5	5	5	5	5	50
96	3	3	3	1	2	1	4	5	2	5	3	32
97	4	4	5	3	2	5	5	5	4	5	5	47
98	4	3	4	4	2	4	4	4	4	5	5	43
99	4	3	4	3	3	3	4	4	4	4	5	41
100	2	2	4	4	2	3	3	3	3	3	4	33
101	3	1	4	2	1	3	4	3	2	4	3	30
102	4	5	4	2	1	4	5	5	4	5	3	42
103	5	3	4	3	3	3	4	3	4	4	3	39
104	4	5	4	2	1	4	5	5	4	5	3	42
105	1	2	2	1	1	3	3	3	4	4	3	27
106	4	5	4	2	1	4	5	5	4	5	3	42
107	3	4	4	3	3	3	3	4	3	3	5	38
108	2	3	4	4	1	3	4	4	2	4	3	34
109	4	3	4	2	3	3	5	4	2	4	3	37
110	3	4	5	4	3	4	4	5	3	4	5	44
111	3	4	5	3	3	3	3	4	4	5	5	42
112	4	4	4	2	3	2	4	4	3	3	4	37

113	3	2	3	1	1	3	3	3	3	4	3	29
114	3	2	3	1	1	1	3	4	1	5	3	27
115	5	3	4	3	4	4	4	3	3	4	4	41
116	3	4	4	3	3	3	4	4	4	4	5	41
117	2	4	4	3	5	5	5	4	4	5	5	46
118	2	2	3	2	2	2	4	4	1	4	3	29
119	4	4	4	4	4	4	4	4	4	4	4	44
120	3	3	4	4	3	3	4	4	2	4	4	38
121	5	4	5	5	4	5	5	5	4	5	5	52
122	3	2	4	3	3	3	4	4	3	4	3	36
123	2	5	3	5	3	3	3	3	3	3	4	37
124	4	3	5	5	4	3	5	4	4	5	5	47
125	4	3	5	5	4	3	5	5	2	5	5	46
126	3	3	3	1	1	3	2	3	1	4	3	27
127	4	3	5	5	2	4	5	4	4	5	5	46
128	4	3	5	3	2	2	4	5	3	4	3	38
129	2	3	4	1	4	2	4	5	2	5	5	37
130	4	2	4	3	3	3	4	3	2	4	3	35
131	4	4	5	1	1	2	5	5	2	5	3	37
132	4	3	4	4	4	4	5	5	3	5	3	44
133	3	5	4	4	5	1	5	4	2	5	3	41
134	2	1	3	2	1	1	4	4	1	4	3	26
135	4	5	5	4	5	1	5	5	1	5	3	43
136	2	4	4	1	1	1	4	5	1	4	3	30
137	3	4	4	2	5	4	2	5	3	5	3	40
138	2	4	3	1	1	1	4	4	1	4	3	28
139	2	1	5	3	2	4	4	3	4	4	3	35
140	3	1	4	2	2	1	4	4	3	4	4	32
141	2	3	4	1	4	2	4	5	2	4	3	34
142	4	2	4	3	3	3	4	3	2	4	3	35
143	4	4	5	1	1	2	5	5	2	5	3	37
144	4	3	4	4	4	4	5	5	4	5	3	45
145	3	5	4	4	5	1	5	5	2	5	3	42
146	2	1	3	2	1	1	4	4	1	4	3	26
147	4	1	5	1	2	1	2	2	1	4	3	26
148	3	5	4	1	1	3	5	3	1	4	3	33
149	2	5	4	1	1	3	5	4	1	5	3	34
150	5	4	5	3	3	2	5	5	1	5	3	41
151	2	5	5	1	4	1	5	5	1	4	5	38
152	5	3	4	2	4	5	5	5	5	5	3	46
153	5	2	2	5	5	5	4	3	4	4	3	42
154	3	4	4	4	4	4	4	3	3	4	5	42
155	4	4	4	4	3	4	4	4	4	5	5	45
156	3	5	5	3	5	5	5	5	3	1	3	43
157	3	3	4	3	5	3	5	5	3	5	3	42
158	1	5	3	1	1	1	5	5	1	4	3	30
159	4	5	4	5	5	4	5	4	4	5	5	50
160	2	2	4	2	2	2	4	2	2	4	3	29
161	3	4	4	4	3	3	4	3	3	4	5	40
162	3	1	5	4	3	3	5	5	4	4	3	40
163	2	4	4	2	4	2	4	4	2	4	3	35
164	3	3	5	3	3	3	5	5	4	5	3	42
165	4	2	2	2	2	2	5	3	2	4	3	31
166	3	4	5	4	5	4	5	4	3	4	3	44
167	3	4	3	3	4	2	5	4	3	4	3	38
168	2	3	4	1	1	1	4	3	1	4	3	27
169	4	2	5	5	2	4	5	5	3	5	3	43
170	3	5	5	5	4	4	5	5	5	5	5	51
171	4	5	4	4	4	4	5	5	5	5	5	50
172	2	2	4	1	2	2	4	3	2	3	5	30
173	4	4	4	3	5	4	5	5	4	5	5	48
174	3	5	4	5	5	4	5	5	4	5	3	48
175	4	5	4	5	5	4	5	4	2	5	3	46

176	3	4	5	4	4	4	5	5	5	5	5	49
177	2	4	5	1	3	1	4	5	1	4	3	33
178	3	2	4	4	4	4	5	5	3	5	3	42
179	3	4	4	2	2	3	4	5	3	4	5	39
180	1	4	3	1	2	1	4	5	2	4	4	31
181	3	3	5	3	3	4	5	5	5	5	5	46
182	3	3	4	3	5	3	4	4	3	5	5	42
183	4	1	5	5	4	4	5	5	5	5	5	48
184	3	3	4	3	4	3	4	3	4	4	5	40
185	4	4	1	4	4	4	5	4	4	4	5	43
186	3	3	4	3	2	3	4	4	2	4	5	37
187	3	2	4	2	1	2	4	4	4	4	3	33
188	2	3	4	2	4	4	4	4	4	4	5	40
189	4	4	5	4	4	3	5	5	2	4	3	43
190	3	3	4	3	3	3	4	4	4	5	3	39
191	3	3	5	2	5	3	5	4	4	5	5	44
192	5	2	2	5	5	5	4	3	4	4	3	42
193	3	4	4	4	4	4	4	3	3	4	5	42
194	4	4	4	4	3	4	5	4	4	4	5	45
195	3	5	5	3	5	5	5	5	3	2	3	44
196	3	3	4	3	5	3	5	5	3	5	3	42
197	1	5	3	1	1	1	4	5	1	4	3	29
198	4	5	4	5	5	4	5	4	4	5	5	50
199	2	2	4	2	2	2	4	2	2	4	3	29
200	3	4	4	4	3	3	4	3	3	4	5	40
201	3	1	5	4	3	3	5	5	4	4	3	40
202	2	4	4	2	4	2	4	4	2	4	3	35
203	3	3	5	3	3	3	5	5	4	5	3	42
204	4	2	2	2	2	2	4	3	2	4	3	30
205	3	4	5	4	5	4	5	4	3	4	3	44
206	3	4	3	3	4	2	5	4	3	4	3	38
207	2	3	4	1	1	1	4	3	1	4	3	27
208	4	2	5	5	2	4	5	5	3	5	3	43
209	3	3	4	2	1	3	5	5	2	5	5	38
210	4	3	4	4	4	4	3	4	3	4	3	40
211	4	3	5	2	2	3	5	5	4	5	5	43
212	3	2	5	2	2	1	4	5	5	5	3	37
213	3	1	3	5	4	5	5	3	2	3	3	37
214	3	3	4	3	3	3	4	3	4	4	5	39
215	5	5	5	3	3	3	3	3	3	4	5	42
216	4	4	4	4	4	4	4	4	3	4	5	44
217	5	4	5	5	5	5	5	5	3	5	5	52
218	3	2	4	4	5	4	5	4	2	4	3	40
219	3	3	4	4	4	4	5	5	4	5	5	46
220	3	2	5	5	5	5	5	5	5	5	5	50
221	4	4	4	4	4	4	3	3	4	4	4	42
222	3	1	3	4	2	1	4	5	3	4	4	34
223	3	2	4	5	2	5	5	4	5	5	5	45
224	3	3	5	3	3	3	4	5	4	5	3	41
225	3	3	4	4	4	4	4	4	3	5	3	41
226	3	3	4	3	4	4	4	2	3	5	5	40
227	3	4	4	2	2	1	4	4	4	4	3	35
228	2	3	4	1	1	1	4	5	3	5	5	34
229	2	2	4	1	1	2	4	4	4	4	3	31
230	4	3	4	4	3	2	4	3	3	4	3	37
231	4	3	4	4	3	3	5	3	3	5	3	40
232	3	4	5	1	1	1	4	4	3	3	3	32
233	4	2	5	3	2	4	5	4	4	4	5	42
234	3	3	4	1	1	2	4	4	3	4	4	33
235	2	3	5	5	2	3	4	4	1	5	5	39
236	2	3	4	3	5	4	4	5	4	5	5	44
237	4	3	3	4	4	4	4	3	4	4	5	42
238	2	4	2	3	3	2	5	5	2	5	3	36

239	3	2	4	1	2	3	5	4	1	4	3	32
240	5	1	5	5	4	5	4	4	4	4	3	44
241	4	2	4	4	3	2	4	4	3	5	5	40
242	4	2	3	4	1	5	5	2	4	5	5	40
243	2	3	5	4	2	2	5	5	4	5	3	40
244	3	4	5	2	2	4	5	5	4	5	3	42
245	3	3	4	3	3	3	4	4	4	4	5	40
246	3	3	3	2	5	4	5	5	4	5	3	42
247	2	4	5	2	3	3	4	5	3	4	3	38
248	3	3	4	2	4	4	4	4	3	4	4	39
249	3	3	5	3	2	3	4	4	4	5	5	41
250	3	3	3	3	3	3	4	4	4	4	5	39
251	4	1	3	5	2	5	5	5	4	5	3	42
252	3	2	5	4	4	3	4	3	5	5	5	43
253	3	1	4	2	3	4	5	5	4	5	5	41
254	4	3	4	1	1	4	4	5	3	4	5	38
255	3	2	4	3	1	2	5	4	4	4	5	37
256	2	3	4	3	5	3	4	4	3	4	4	39
257	3	2	4	1	2	4	4	4	4	4	3	35
258	3	2	4	3	1	2	4	4	1	4	5	33
259	3	4	5	2	2	2	4	5	2	4	5	38
260	2	1	5	1	2	2	5	4	2	4	5	33
261	3	2	4	2	3	4	5	4	4	5	5	41
262	4	4	3	4	4	4	4	5	4	5	3	44
263	3	3	4	2	4	1	5	4	3	4	3	36
264	3	1	4	3	2	3	5	4	4	4	5	38
265	4	1	5	5	5	4	5	4	1	5	3	42
266	4	3	5	4	2	3	5	4	1	4	3	38
267	2	3	4	2	3	3	5	4	3	4	5	38
268	3	2	5	3	4	4	5	4	3	5	5	43
269	3	1	5	1	4	1	4	4	3	3	4	33
270	2	3	4	3	4	2	4	2	2	3	3	32
271	4	3	5	2	1	2	5	1	3	4	5	35
272	4	3	4	2	3	4	4	2	4	5	5	40
273	2	2	5	1	1	1	3	1	5	4	3	28
274	3	3	2	1	1	3	3	2	2	3	3	26
275	5	2	5	4	3	4	4	5	5	5	5	47
276	2	5	4	2	4	4	3	4	1	4	3	36
277	3	1	5	1	1	3	4	4	1	4	3	30
278	5	2	4	3	3	3	4	5	3	4	3	39
279	4	4	4	2	1	4	5	5	2	4	5	40
280	1	2	1	1	1	3	3	2	1	4	4	23
281	4	4	5	2	1	4	5	4	3	5	3	40
282	3	2	4	3	3	3	3	3	2	4	5	35
283	2	1	3	4	1	3	4	4	1	3	3	29
284	4	2	4	2	3	3	5	4	2	4	4	37
285	3	4	5	4	3	4	4	5	3	5	5	45
286	3	4	5	3	3	3	3	3	1	4	5	37
287	4	2	4	2	3	2	4	1	1	3	3	29
288	3	1	3	1	1	3	3	1	1	2	3	22
289	3	1	3	1	1	1	3	1	1	4	3	22
290	5	5	5	3	4	4	4	4	3	5	5	47
291	3	4	4	3	3	3	4	4	3	3	5	39
292	2	5	5	3	5	5	5	4	4	5	5	48
293	2	1	3	2	2	2	4	3	1	2	3	25
294	4	5	5	4	4	4	4	5	4	5	5	49
295	3	3	3	4	3	3	4	4	1	3	3	34
296	5	5	5	5	4	5	5	5	4	5	5	53
297	3	2	4	3	3	3	4	4	2	5	3	36
298	2	4	4	5	3	3	3	5	3	5	5	42
299	4	5	5	5	4	3	5	5	4	5	5	50
300	4	5	5	5	4	3	5	5	2	5	5	48
301	3	4	3	1	1	3	2	1	1	3	3	25

302	4	5	5	5	2	4	5	5	4	5	5	49
303	4	4	4	3	2	2	4	1	1	3	3	31
304	3	4	4	2	3	4	4	5	4	5	3	41
305	3	4	5	4	3	4	5	5	4	5	5	47
306	4	5	4	5	1	5	4	4	5	5	4	46
307	3	5	5	5	3	4	5	5	4	5	5	49
308	4	5	4	5	4	5	5	5	3	5	5	50
309	3	2	4	2	3	3	4	2	1	3	3	30
310	1	2	4	1	1	1	3	1	1	4	3	22
311	3	4	5	4	5	4	5	3	3	5	3	44
312	4	4	5	5	5	4	4	5	4	5	5	50
313	3	5	4	5	5	4	4	2	2	5	5	44
314	4	5	5	5	4	4	5	5	4	5	5	51
315	4	5	5	4	4	4	4	2	4	5	5	46
316	3	1	4	3	4	3	5	4	2	4	4	37
317	2	5	3	5	2	4	4	3	2	4	3	37
318	3	3	1	3	2	3	3	4	2	3	3	30
319	3	3	5	4	1	3	3	1	1	3	3	30
320	4	5	5	4	5	4	4	5	3	5	3	47
321	4	5	5	4	5	3	4	4	4	5	3	46
322	4	4	5	4	5	4	5	5	4	5	5	50
323	4	5	5	5	5	5	5	5	4	5	4	52
324	2	1	4	3	5	3	3	5	4	4	3	37
325	4	5	5	4	3	5	5	5	4	5	5	50
326	3	5	4	4	5	4	4	2	3	5	3	42
327	4	5	5	3	2	2	4	2	3	3	4	37
328	2	4	4	3	3	3	3	3	3	4	4	36
329	2	2	5	3	3	3	4	4	3	4	3	36
330	3	3	5	3	4	3	5	4	2	5	5	42
331	3	1	4	4	3	4	5	4	5	5	5	43
332	3	1	4	1	1	3	4	4	3	4	5	33
333	3	3	4	3	3	3	5	5	3	4	5	41
334	4	3	5	5	5	5	4	4	4	4	5	48
335	4	3	4	4	5	4	4	4	4	5	3	44
336	4	3	4	3	3	3	4	4	4	5	4	41
337	3	3	5	3	3	3	4	4	3	5	4	40
338	4	3	4	4	4	4	4	4	4	5	3	43
339	4	4	5	4	4	4	4	4	5	5	5	48
340	4	4	4	4	4	4	4	4	3	5	5	45
341	3	5	4	3	3	3	3	5	3	4	5	41
342	3	3	4	4	2	4	4	3	4	5	3	39
343	3	5	5	2	4	4	4	4	2	5	5	43
344	3	1	1	2	2	3	1	1	3	1	3	21
345	4	4	4	4	2	4	4	5	5	4	5	45
346	4	2	4	3	2	4	4	2	4	4	3	36
347	3	4	5	4	3	4	4	4	2	4	3	40
348	3	3	3	4	3	3	4	4	4	4	3	38
349	4	5	4	4	4	3	4	4	3	4	3	42
350	3	2	4	3	2	1	3	4	4	4	3	33
351	4	3	2	4	4	4	5	3	1	4	3	37
352	4	5	4	5	5	5	5	5	4	5	3	50
353	4	2	4	5	5	4	4	5	5	5	5	48
354	3	1	4	4	4	4	4	4	2	4	3	37
355	4	5	5	5	4	4	5	5	4	5	5	51
356	3	4	4	4	3	2	5	4	4	4	5	42
357	4	3	5	4	5	3	5	4	4	5	3	45
358	4	4	5	4	4	3	5	4	3	4	5	45
359	5	3	5	5	4	5	4	3	4	3	3	44
360	5	1	3	3	3	3	3	3	3	2	4	33
361	3	2	5	3	3	5	5	5	5	5	4	45
362	3	4	4	4	4	4	5	4	4	4	5	45
363	3	4	3	4	4	4	5	5	4	5	5	46
364	2	4	4	4	3	4	5	3	4	5	5	43

365	5	4	5	4	5	5	5	5	4	5	5	52
366	4	3	4	4	5	5	5	5	4	5	3	47
367	3	1	5	4	2	4	5	5	4	4	5	42
368	3	2	5	2	1	1	4	3	4	5	3	33
369	2	4	4	1	1	3	4	4	3	4	3	33
370	4	4	5	5	4	3	5	5	3	5	3	46
371	2	1	3	3	1	1	3	3	2	2	3	24
372	3	4	4	2	3	5	4	4	4	5	5	43
373	3	5	5	4	3	5	5	5	5	5	5	50
374	3	1	4	5	1	4	4	5	5	5	5	42
375	3	4	5	4	3	5	5	5	3	5	5	47
376	4	4	4	5	4	5	5	3	3	5	4	46
377	3	3	4	3	3	3	4	3	2	3	3	34
378	1	3	4	1	2	1	3	2	1	4	3	25
379	3	5	5	4	4	5	5	3	3	5	3	45
380	4	4	5	5	4	4	4	4	4	5	3	46
381	3	4	4	5	3	3	4	2	2	4	5	39
382	4	2	5	5	5	5	5	5	4	5	5	50
383	4	5	5	4	3	5	4	3	4	5	5	47
384	3	2	4	3	3	2	5	5	3	4	3	37
385	2	3	3	5	4	3	4	3	3	3	3	36
386	3	1	4	3	2	2	3	4	3	2	5	32
387	4	4	5	4	1	3	3	4	2	5	3	38
388	3	4	5	4	3	4	4	4	3	5	4	43
389	4	3	4	4	3	3	4	4	4	5	3	41
390	4	5	5	4	3	4	5	4	4	5	4	47
391	4	5	5	5	4	5	5	5	4	5	5	52
392	2	1	4	3	1	2	3	4	4	3	3	30
393	4	2	4	4	2	4	5	4	4	5	5	43
394	3	3	3	4	3	3	4	4	3	4	3	37
395	4	3	5	3	1	2	4	4	4	3	3	36
396	3	2	4	3	2	1	3	3	3	3	4	31
397	4	2	4	3	3	3	4	5	4	3	3	38
398	3	3	5	3	3	3	5	5	2	5	5	42
<b>VARIANZA</b>	0,7157	1,4077	0,6255	1,6391	1,7122	1,227	0,5372	0,9454	1,2901	0,606399	0,883513	

### Fórmula para el alpha de cronbach

$$\alpha = \frac{K}{K-1} \left[ 1 - \frac{\sum S_i^2}{S_T^2} \right]$$

$\alpha$  Coeficiente de confiabilidad = 0,80025343 → 80,0%

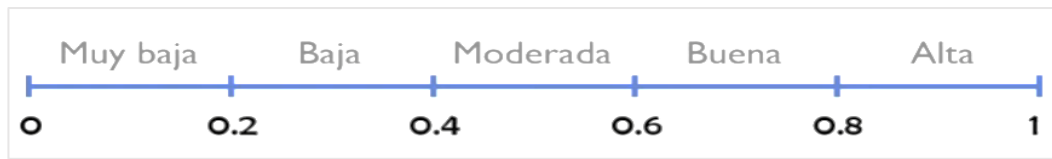
k Número de ítems del instrumento = 11

$\sum S_i^2$  Sumatoria de las varianzas de los ítems = 11,5897515

$S_T^2$  Varianza total del instrumento = 42,5316848



## Línea de referencia para el valor del alpha de Cronbach



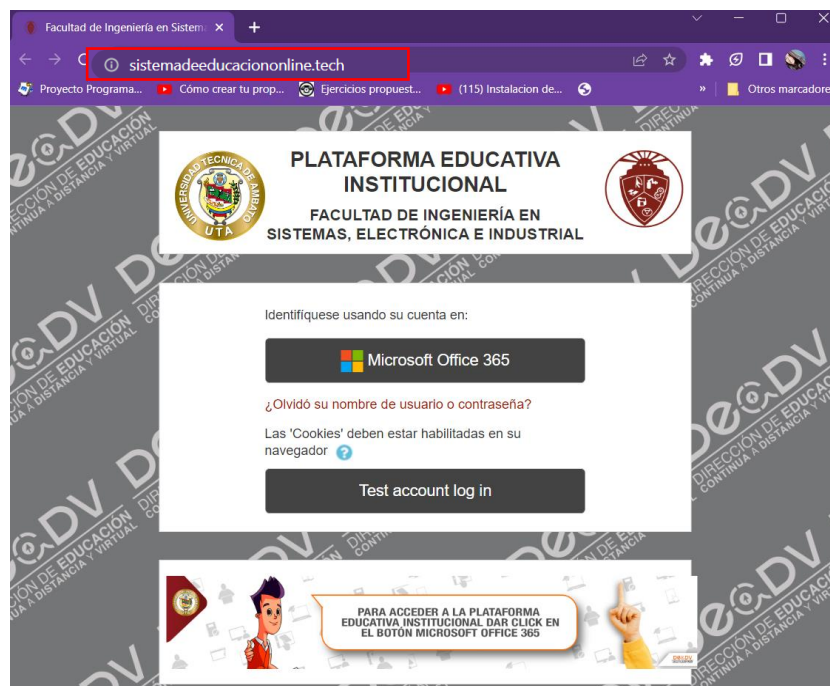
**Figura 26.** Valor del alpha de cronbach

**Elaborado por:** Fabiana Jaramillo

## Anexo B

### Plataforma educativa de la FISEI

- **Dominio ofuscado:** sistemadeeducaciononline.tech

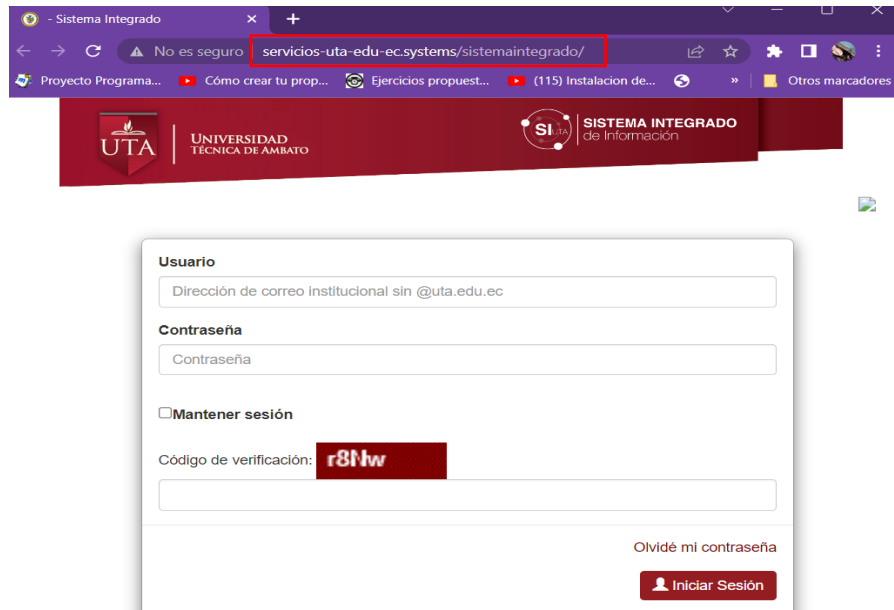


**Figura 27.** Página clonada de la plataforma educativa de la FISEI

**Elaborado por:** Fabiana Jaramillo

## Sistema integrado de la UTA

- **Dominio ofuscado:** servicios-uta-edu-ec.systems

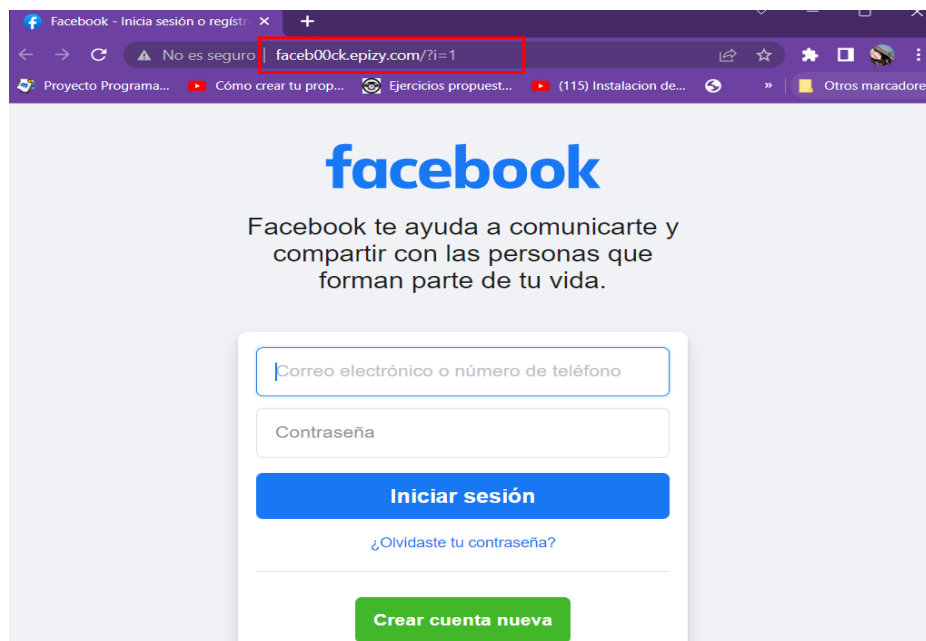


**Figura 28.** Página clonada del sistema integrado UTA

**Elaborado por:** Fabiana Jaramillo

## Facebook

- **Dominio ofuscado:** faceb00ck.epizy.com

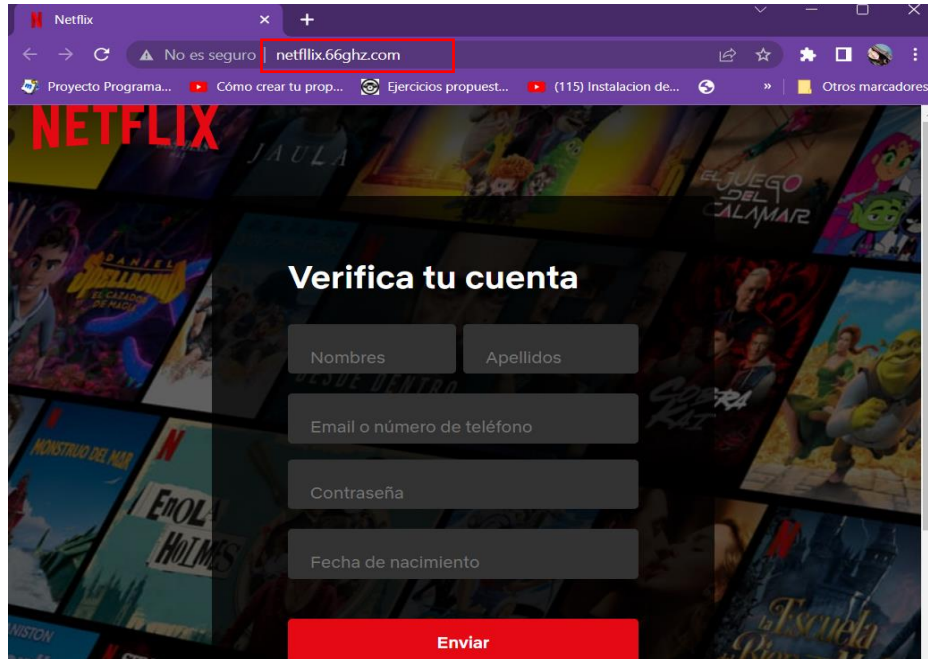


**Figura 29.** Página clonada de facebook

**Elaborado por:** Fabiana Jaramillo

## Netflix

- **Dominio ofuscado:** faceb00ck.epizy.com

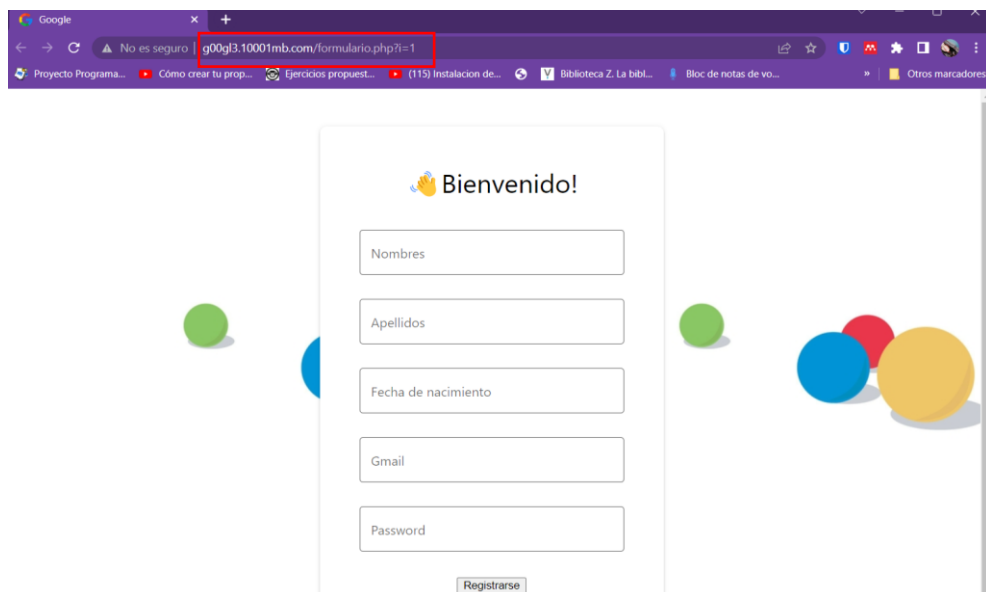


**Figura 30.** Página clonada de Netflix

**Elaborado por:** Fabiana Jaramillo

## Google

- **Dominio ofuscado para 85nicod:** g00g13.10001mb.com

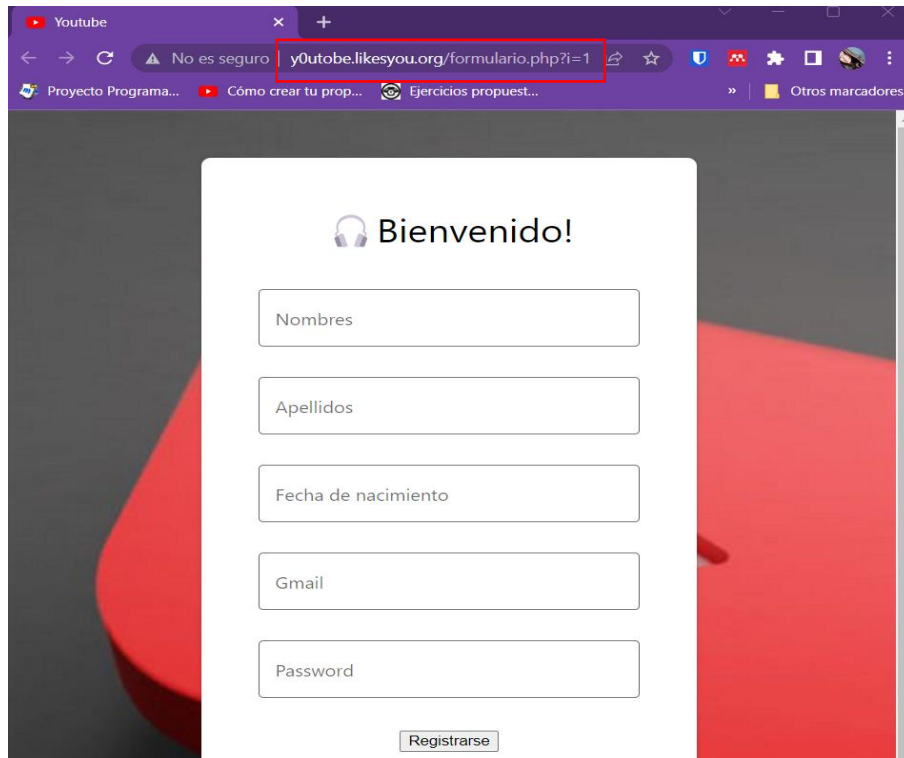


**Figura 31.** Formulario de engaño de parte de Google

**Elaborado por:** Fabiana Jaramillo

## YouTube

- **Dominio ofuscado para youtube:** y0utobe.likesyou.org



**Figura 32.** Formulario de engaño de parte de youtube

**Elaborado por:** Fabiana Jaramillo

## Anexo C

### Funciones para la extracción de las características

```
def url_tiene_ip(url):
    # Buscar un patrón de dirección IP en la URL
    match = re.search(r"\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b", url)
    if match:
        # Si se encuentra una coincidencia, la URL tiene una dirección IP
        return 2
    else:
        # Si no se encuentra una coincidencia, la URL no tiene una dirección IP
        return 1

def url_tiene_ssl(url):
    # Analizar la estructura de la URL
    parsed_url = urllib.parse.urlparse(url)
    # Comprobar si el esquema de la URL es "https"
    if parsed_url.scheme == "https":
        return 2
```

```

else:
    return 1

def num_puntos(url):
    # Contar el número de puntos en la URL
    dots = url.count(".")
    # Si el número de guiones es 0, devolver 0.001 en su lugar
    if dots == 0:
        dots = 0.001
    return dots

def long_host(url):
    # Analizar la estructura de la URL
    parsed_url = urllib.parse.urlparse(url)
    # Obtener el host de la URL
    host = parsed_url.hostname
    # Calcular la longitud del host
    host_length = len(host)
    return host_length

def long_ruta(url):
    # Analizar la estructura de la URL
    parsed_url = urllib.parse.urlparse(url)
    # Obtener la ruta del archivo de la URL
    path = parsed_url.path
    # Si la consulta es vacía, devuelve 0.001
    if path == "":
        return 0.001
    # En caso contrario, devuelve la longitud de la ruta del archivo
    else:
        return len(path)

def long_consulta(url):
    # Analizar la estructura de la URL
    parsed_url = urllib.parse.urlparse(url)
    # Obtener la consulta de la URL
    query = parsed_url.query
    # Si la consulta es vacía, devuelve 0.001
    if query == "":
        return 0.001
    # En caso contrario, devuelve la longitud de la consulta
    else:
        return len(query)

def long_total(url):
    # Obtener la longitud total de la URL
    total_length = len(url)
    return total_length

```

```

def 88nicode88_alfabetica(url):
    # Analizar la estructura de la URL y obtener el host
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Inicializar el diccionario de ocurrencias
    occurrences = {}
    # Recorrer el host y contar el número de ocurrencias de cada carácter
    for char in host:
        if char in 88nicode88ca88:
            88nicode88ca88[char] += 1
        else:
            88nicode88ca88[char] = 1
    # Inicializar la entropía en 0
    entropy = 0
    # Recorrer el diccionario de ocurrencias y aplicar la fórmula de la entropía
    for char, count in 88nicode88ca88.items():
        probability = count / len(host)
        entropy -= probability * math.log2(probability)

    # Truncar el resultado a cinco decimales
    entropy = round(entropy, 5)
    return entropy

def tasa_conti_carac(url):
    # Analizar la estructura de la URL y obtener el host
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Inicializar el contador de caracteres consecutivos en 0
    consecutive_chars = 0
    # Recorrer el host y contar los caracteres consecutivos
    for i in range(1, len(host)):
        if host[i] == host[i-1]:
            consecutive_chars += 1
    # Calcular la tasa de continuidad
    continuity_rate = consecutive_chars / len(host)
    # Si la tasa de continuidad es 0.0, devolver 0.001 en su lugar
    if continuity_rate == 0.0:
        continuity_rate = 0.001
    # Truncar el resultado a cinco decimales
    continuity_rate = round(continuity_rate, 5)
    return continuity_rate

def num_guiones(url):
    # Analizar la estructura de la URL y obtener el host
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Contar el número de guiones en el host
    num_dashes = host.count("-")

```

```

# Si el número de guiones es 0, devolver 0.001 en su lugar
if num_dashes == 0:
    num_dashes = 0.001
return num_dashes

def num_carac_esp(url):
    # Analizar la estructura de la URL y obtener el host
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Usar una expresión regular para contar el número de caracteres especiales
    num_special_chars = len(re.findall(r'^\w', host))
    # Si el número de caracteres especiales es 0, devolver 0.001 en su lugar
    if num_special_chars == 0:
        num_special_chars = 0.001
    return num_special_chars

def patron_ldl_or_dld (url):
    # Analizar la estructura de la URL y obtener el hostname
    parsed_url = urllib.parse.urlparse(url)
    hostname = parsed_url.hostname
    # Buscar el patrón "letra-dígito-letra" o "dígito-letra-dígito" en el hostname
    ldl_pattern = r'[a-z][0-9][a-z]'
    dld_pattern = r'[0-9][a-z][0-9]'
    if re.search(ldl_pattern, hostname) or re.search(dld_pattern, hostname):
        return 2
    else:
        return 1

def num_arrobas(url):
    # Buscar el símbolo @ en la URL
    at_pattern = r'@'
    ats = re.findall(at_pattern, url)
    # Devolver el número de símbolos @ encontrados o 0.001 si no se encontró
    ninguno
    return len(ats) if ats else 0.001

def num_letras(url):
    # Analizar la estructura de la URL y obtener el host
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Usar una expresión regular para contar el número de letras en el host
    letter_pattern = r'[a-zA-Z]'
    letters = re.findall(letter_pattern, host)
    # Devolver el número de letras encontradas o 0.001 si no se encontró ninguno
    return len(letters) if letters else 0.001

def num_digitos(url):
    # Analizar la estructura de la URL y obtener el host

```

```

    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Usar una expresión regular para contar el número de dígitos en el host
    digit_pattern = r'\d'
    digits = re.findall(digit_pattern, host)
    # Devolver el número de dígitos encontrados o 0.001 si no se encontró ninguno
    return len(digits) if digits else 0.001

def num_guionesbajos(url):
    # Analizar la estructura de la URL y obtener el host
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Usar una expresión regular para contar el número de guiones bajos en el host
    underscore_pattern = r'_'
    underscores = re.findall(underscore_pattern, host)
    # Devolver el número de guiones bajos encontrados o 0.001 si no se encontró
ninguno
    return len(underscores) if underscores else 0.001

def url_tiene_www(url):
    # Analizar la estructura de la URL y obtener el host
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Comprobar si el host comienza con "www"
    if host.startswith("www"):
        return 2
    else:
        return 1

def num_palabras_sospechosas(url):
    # Analizar la estructura de la URL y obtener el host y la ruta del archivo
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    path = parsed_url.path
    # Definir una lista de palabras sospechosas
    suspicious_words = ["include", "signin", "login", "ebay", "account", "secure",
"confirm", "bank", "logon", "cmd", "admin", "paypal"]
    # Contar el número de palabras sospechosas en el host y en la ruta del archivo
    count = 0
    for word in suspicious_words:
        count += host.count(word)
        count += path.count(word)
    # Si no se encontró ninguna palabra sospechosa, devolver 0.001
    if count == 0:
        return 0.001
    # Si se encontró al menos una palabra sospechosa, devolver el número de
palabras sospechosas encontradas
    else:

```



```

        return count

def num_tld(url):
    #Extrae el TLD y cuenta el número de puntos después del TLD en una URL dada.
    # Utilizar el método urlsplit para dividir la URL en sus componentes
    91nico_url = urlsplit(url)
    # Obtener el dominio de la URL
    domain = 91nico_url.netloc
    # Dividir el dominio en sus partes
    domain_parts = domain.split('.')
    # Obtener el último segmento del dominio (que es el TLD)
    tld = domain_parts[-1]
    # Contar el número de puntos después del TLD
    subdomain_count = len(domain_parts) - 1
    return (subdomain_count)

def url_host_hex_or_base64(url):
    # Analizar la estructura de la URL y obtener el host
    parsed_url = urllib.parse.urlparse(url)
    host = parsed_url.hostname
    # Usar una expresión regular para verificar si el host está codificado en
    hexadecimal o base64
    is_encoded = bool(re.search(r'^[0-9a-fA-F]+$|^[A-Za-z0-9+/=]+$' , host))
    resultado = 2 if is_encoded == True else 1
    return resultado

def url_path_hex(url):
    # Analizar la estructura de la URL y obtener la ruta
    parsed_url = urllib.parse.urlparse(url)
    path = parsed_url.path
    # Usar una expresión regular para buscar la presencia de caracteres codificados
    en forma hexadecimal
    hex_encoding_pattern = re.compile(r'%[0-9a-fA-F]{2}')
    hex = bool(hex_encoding_pattern.search(path))
    resultado = 2 if hex == True else 1
    return resultado

def url_tiene_unicode(url):
    # Usar una expresión regular para buscar cualquier secuencia de caracteres que
    comience con %u
    match = re.search(r'%u[0-9a-fA-F]{4}', url)
    if match:
        # Si se encontró una coincidencia, devolver 2
        return 2
    else:
        # Si no se encontró ninguna coincidencia, devolver 1
        return 1

```

```

def url_ejecutable(url):
    # Analizar la estructura de la URL y obtener el nombre del archivo
    parsed_url = urllib.parse.urlparse(url)
    file_name = parsed_url.path.split('/')[-1]
    # Comparar el nombre del archivo con una lista de extensiones de archivos
    ejecutables
    executable_extensions = [".exe", ".com", ".bat"]
    for extension in executable_extensions:
        if file_name.endswith(extension):
            return 2
    # Si no se encontró ninguna coincidencia, devolver 1
    return 1

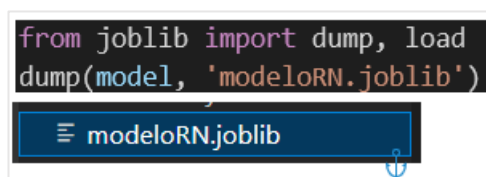
def url_ofus_redirec(url):
    # Usar una expresión regular para buscar cualquier secuencia de caracteres que
    comience con @
    match = re.search(r'^@[^@]*$', url)
    if match:
        # Si se encontró una coincidencia, devolver 2
        return 2
    else:
        # Si no se encontró ninguna coincidencia, devolver 1
        return 1

```

**Figura 33.** Funciones para la extracción de las características

**Elaborado por:** Fabiana Jaramillo

## Anexo D



```

from joblib import dump, load
dump(model, 'modeloRN.joblib')

```

≡ modeloRN.joblib

**Figura 34.** Guardado del modelo RNA con la librería joblib

**Elaborado por:** Fabiana Jaramillo

## Anexo E

```

def url_has_vector(url):
    # Obtener el resultado de la función url_has_ip
    result1 = url_tiene_ip(url)
    result2 = url_tiene_ssl(url)
    result3 = num_puntos(url)
    result4 = long_host(url)
    result5 = long_ruta(url)

```

```

result6 = long_consulta(url)
result7 = long_total(url)
result8 = 93nicode93_alfabetica(url)
result9 = tasa_conti_carac(url)
result10 = num_guiones(url)
result11 = num_carac_esp(url)
result12 = patron_ldl_or_dld(url)
result13 = num_arrobas(url)
result14 = num_letras(url)
result15 = num_digitos(url)
result16 = num_guionesbajos(url)
result17 = url_tiene_www(url)
result18 = num_palabras_sospechosas(url)
result19 = num_tld(url)
result20 = url_host_hex_or_base64(url)
result21 = url_ruta_hex(url)
result22 = url_tiene_unicode(url)
result23 = url_ejecutable(url)
result24 = url_ofus_redirec(url)
# Crear una lista con el resultado
vector = [result1, result2, result3, result4, result5, result6, result7,
result8, result9, result10, result11, result12,
result13, result14, result15, result16, result17, result18, result19, result20,
result21, result22, result23, result24]
return vector

```

**Figura 35.** Función url\_has\_vector que retorna el vector de características de la URL

**Elaborado por:** Fabiana Jaramillo

```

def predecir (url):

    model = load('modeloRN.joblib')

    vector = url_has_vector(url)

    prediccion = model.predict([vector], verbose=0)
    predicted_class = np.argmax(prediccion, axis=1)

    return predicted_class

```

**Figura 36.** Función predecir que retorna la etiqueta predicha

**Elaborado por:** Fabiana Jaramillo

## Anexo F

```
modelo.py  app.py  x
app.py > health
1  from flask import Flask, request, json
2  from modelo import predecir
3  from flask_cors import CORS
4
5  app = Flask(__name__)
6  CORS(app)
7
8  @app.route("/rna" ,methods = ['GET'])
9
10 def hello_world():
11
12     url = request.args.get('url')
13
14     y = predecir(url)
15     respuesta = "Legitima"
16     if str(y[0]) == "2":
17         respuesta = "Phishing"
18     response = app.response_class(
19         response=json.dumps(respuesta),
20         status=200,
21         mimetype='application/json'
22     )
23
24     return response
```

**Figura 37.** API RESTful utilizando Flask

**Elaborado por:** Fabiana Jaramillo

## Anexo G

```
popup.html  {} manifest.json  JS content-scripts.js  x
JS content-scripts.js > ...
1  url = window.location.href;
2
3  console.log("url: " + url)
4
5  response = fetch(`http://localhost:5000/rna?url=${url}`)
6  .then(response => response.json())
7  .then(data => {
8      console.log("data: " + data);
9
10     if (data == "Legitima") {
11         console.log("Funciona, es una página legitima");
12     } else {
13         alert('Esta página parece ser Phishing te recomiendo salir')
14     }
15 }
16 .catch(error => {
17     console.log(error);
18 });
19
20 console.log(response)
```

**Figura 38.** Archivo conten-scripts.js de la extensión para Chrome

**Elaborado por:** Fabiana Jaramillo

## Anexo H

```
popup.html  {} manifest.json  JS content-scripts.js
manifest.json > ...
1  {
2    "name": "Phish Alert",
3    "description": "Esta extensión identifica una página web Phishing y muestra una alerta al usuario.",
4    "version": "1.0",
5    "manifest_version": 3,
6    "action": {
7      "default_popup": "popup.html",
8      "default_icon": {
9        "16": "/imagenes/PhishingAlert16.png",
10       "32": "/imagenes/PhishingAlert32.png",
11       "48": "/imagenes/PhishingAlert48.png",
12       "128": "/imagenes/PhishingAlert128.png"
13     }
14   },
15   "icons": {
16     "16": "/imagenes/PhishingAlert16.png",
17     "32": "/imagenes/PhishingAlert32.png",
18     "48": "/imagenes/PhishingAlert48.png",
19     "128": "/imagenes/PhishingAlert128.png"
20   },
21   "content_scripts": [
22     {
23       "matches": ["*://*/*"],
24       "js": ["content-scripts.js"]
25     }
26   ]
27 }
```

**Figura 39.** Archivo manifest.json de la extensión de Chrome

**Elaborado por:** Fabiana Jaramillo



**Figura 40.** Finalización de la lista de tareas

**Elaborado por:** Fabiana Jaramillo