

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Desarrollo de asistente virtual para el acompañamiento en la navegación de sitios web o descarga de aplicaciones de forma segura

Trabajo de graduación presentado por Diana Zaray Corado López para optar al grado académico de Licenciada en Ingeniería en Ciencia de la Computación y Tecnología de la Información

Guatemala,

2023

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Desarrollo de asistente virtual para el acompañamiento en la navegación de sitios web o descarga de aplicaciones de forma segura

Trabajo de graduación presentado por Diana Zaray Corado López para optar al grado académico de Licenciado en Ingeniería en Ciencia de la Computación y Tecnología de la Información

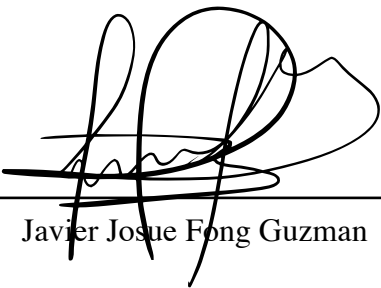
Guatemala,

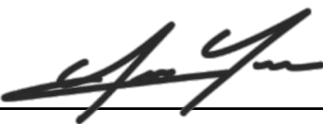
2023


Vo.Bo.:

(f) 
Javier Josue Fong Guzman

Tribunal Examinador:

(f) 
Javier Josue Fong Guzman

(f) 
Jorge Andres Yass Coy

(f) 
Douglas Leonel Barrios Gonzalez

Fecha de aprobación: Guatemala, 04 de diciembre de 2023.

Lista de figuras	VII
Lista de cuadros	VIII
Resumen	IX
Abstract	X
1. Introducción	1
2. Justificación	2
3. Objetivos	4
3.1. Objetivo general	4
3.2. Objetivos específicos	4
4. Alcance	5
5. Marco teórico	6
5.1. Asistente virtual	6
5.1.1. ¿Cómo funciona?	6
5.1.2. Tipos	7
5.1.3. Relevancia en la educación	7
5.2. <i>Natural language processing</i>	8
5.2.1. <i>Natural language understanding</i>	8
5.2.2. <i>Dialog management</i>	8
5.2.3. <i>Natural language generation</i>	9
5.3. <i>Large language models</i> (LLM)	9
5.3.1. Entrenamiento de los LLM	10
5.4. Inteligencia artificial	10
5.4.1. Tipos de inteligencia	10
5.4.2. Habilidades de IA	12
5.5. Redes neuronales	12
5.5.1. ¿Cómo funciona?	12

5.5.2. Arquitecturas	14
5.5.3. Perceptrón multicapa	14
5.5.4. Redes neuronales recurrentes (RNN)	14
5.5.5. <i>Generative adversarial nets</i> (GAN)	15
5.5.6. <i>Transformer</i>	16
5.5.7. <i>Generative pre-trained transformer</i> (GPT)	17
5.6. Seguridad de la información	18
5.6.1. Características de la información	19
5.6.2. Ciberseguridad	19
5.6.3. El factor humano de la ciberseguridad	19
5.7. Psicología y ciberseguridad	20
5.7.1. Sistemas de pensamiento	20
5.7.2. Sesgos en ciberseguridad	21
6. Responsible AI	23
6.1. Datos	23
6.2. Limitaciones	23
7. Metodología	25
7.1. Selección de modelo de LLM	25
7.2. Implementación de modelo LLM	27
7.3. Generación de datos de entrenamiento	28
7.3.1. Conjunto de datos de entrenamiento para GPT-3	30
7.3.2. Conjunto de entrenamiento para GPT-3.5	30
7.4. <i>Fine-tuning</i>	31
7.5. Desarrollo de modelo de clasificación de dominios	33
7.6. Pruebas de conocimiento e interacción con el asistente virtual	37
7.7. Simulación de ataque de <i>phishing</i>	38
8. Resultados	41
8.1. Pruebas de prototipo	41
8.2. Modelo especializado	42
8.3. Modelo de clasificación de dominios	45
8.4. Pruebas de conocimiento e interacción con el asistente virtual	48
8.4.1. Pruebas preliminares	48
8.4.2. Pruebas piloto	49
8.5. Simulación de ataque de <i>phishing</i>	52
8.5.1. Prueba preliminar	52
8.5.2. Prueba piloto	54
9. Conclusiones	56
10.Recomendaciones	57
11.Bibliografía	58

12. Anexos	63
12.1. Datos de modelo de clasificación de dominios	63
12.2. Consentimiento informado	63
12.3. Formulario de pruebas para evaluación de aceptación de modelos LLM	65
12.4. Cuestionario de conocimiento en ciberseguridad de prueba piloto	66
12.5. Aplicación	69
12.6. Test de significancia para los resultados de la simulación de ataque de <i>phishing</i>	70
12.6.1. Clics en enlaces maliciosos	70
12.6.2. Navegación en páginas de <i>phishing</i>	70
12.7. Entrenamiento de modelo de clasificación de dominios con un mayor número de datos de <i>ransomware</i>	71

Lista de figuras

1. Componentes básicos de una neurona biológica	13
2. Arquitectura de un perceptrón	15
3. Arquitectura base de una red neuronal recurrente	16
4. Arquitectura simple de GPT	18
5. Cotización de uso de instancia en Amazon EC2	26
6. Costos por uso de modelo GPT-3.5 de OpenAI	27
7. Costo de proceso de fine tuning en modelos de OpenAI	27
8. Estimación de costos de uso del modelo de OpenAI	27
10. Implementación de dos modelos de GPT-3. Respuesta rápida text-davinci-002 y res- puesta lenta text-davinci-003	29
11. Comparación de respuesta de dos modelos de GPT-3 al consultar sobre un ataque profesional	29
9. Desarrollo de un web scraper mediante Python	29
12. Ejemplo del formato de datos de entrenamiento para modelos de GPT-3	30
13. Ejemplo de conjunto de datos de entrenamiento	30
14. Ejemplo del formato de datos de entrenamiento para modelos de GPT-3.5	30
15. Snippet de conjunto de datos utilizado para hacer fine tuning del modelo	31
16. Resultado del proceso de fine tuning al modelo de completación de chat de OpenAI	33
17. Distribución de datos de entrenamiento para modelo de clasificación de dominios	34
18. Promedio de tiempo activo de un dominio	36
19. Porcentaje de símbolos promedio en cada uno de los tipos de dominio	36
20. Cantidad promedio de letras consecutivas para cada tipo de dominio	36
21. Arquitectura del modelo de clasificación de dominios	37
22. Imitación de página de inicio de sesión de Spotify	39
23. Copia de página de login de Facebook para prueba de simulación de phishing	39
24. Copia de correo electrónico enviado por Facebook al momento de cambiar contraseña	40
25. Solicitud de un dato interesante de ciberseguridad al modelo GPT-3.5 de OpenAI	42
26. Solicitud de un dato interesante sobre ciberseguridad al modelo fine tune	42
27. Solicitud de información sobre un ataque mediante el modelo de GPT-3.5	43
28. Solicitud de información sobre un ataque mediante el modelo fine tuned	44

29. <i>Solicitud de información sobre un ataque mediante el modelo fine tuned con datos de entrenamiento formateados</i>	45
30. <i>Métricas durante el proceso de entrenamiento del modelo de clasificación de dominios</i>	46
31. <i>Curva de aprendizaje de entrenamiento de modelo de detección de dominios maliciosos</i>	46
32. <i>Reporte de clasificación del modelo de clasificación de dominios con el conjunto de prueba</i>	47
33. <i>Matriz de confusión del modelo de clasificación de dominios</i>	47
34. <i>Gráfico de respuestas a cuestionario de ciberseguridad antes de utilizar ELA en prueba preliminar</i>	49
35. <i>Gráfica de respuestas al cuestionario de ciberseguridad antes de usar ELA en prueba piloto</i>	51
36. <i>Gráfica de respuestas de ciberseguridad después de usar ELA</i>	51
37. <i>Gráfico de interacción de usuarios con el correo de simulación de phishing</i>	53
38. <i>Gráfico de interacción con el ataque de phishing mediante mensaje de texto</i>	54
39. <i>Consentimiento informado</i>	64
40. <i>Capturas de pantalla de la aplicación ELA</i>	69
42. <i>Curva de aprendizaje de entrenamiento de modelo de detección de dominios maliciosos con incremento en datos de ransomware y modificación en épocas de entrenamiento y tasa de aprendizaje</i>	72
41. <i>Curva de aprendizaje de entrenamiento de modelo de detección de dominios maliciosos con incremento en datos de ransomware</i>	72
43. <i>Reporte de clasificación de modelo de detección de dominios maliciosos con incremento en datos de ransomware y modificación en épocas de entrenamiento y tasa de aprendizaje</i>	73

Lista de cuadros

1.	<i>Características utilizadas para entrenamiento del modelo de clasificación de dominios</i>	35
2.	<i>Asociación entre número de pregunta en gráfica y pregunta en cuestionario para prueba preliminar</i>	49
3.	<i>Asociación entre número de pregunta en gráfica y pregunta en cuestionario para prueba piloto</i>	50
4.	<i>Relación de la protección de los dos tipos de versiones de ELA con el acceso al link de Facebook en ataque de smishing</i>	55
5.	<i>Tabla de contingencia de clics en simulación de ataques de phishing</i>	70
6.	<i>Tabla de contingencia para el ingreso a una página de phishing</i>	71

En la actualidad, las tecnologías tradicionales han mostrado limitaciones en la protección contra ataques cibernéticos. Aunque los antivirus pueden reconocer patrones o características para identificar y detectar amenazas, estos se han vuelto insuficientes frente a los nuevos vectores de ataque que explotan sesgos cognitivos o vulnerabilidades de los usuarios. En respuesta a esta necesidad, se ha desarrollado un asistente virtual con la capacidad de proporcionar educación en ciberseguridad, cuyo objetivo principal es que los usuarios puedan identificar cuándo están expuestos a un posible ataque. Para ofrecer un acompañamiento más efectivo, se creó un modelo de detección de dominios maliciosos que logra clasificar correctamente un dominio en un 70 % de las ocasiones. Finalmente, se ha demostrado que, al utilizar un asistente virtual y recibir notificaciones sobre el tipo de ataque al que se podría estar expuesto al acceder a un dominio malicioso, la probabilidad de que una persona se proteja exitosamente aumenta en un 50 %. Estos resultados resaltan la importancia de adoptar enfoques proactivos y educativos para mejorar la conciencia y la seguridad cibernética de los usuarios en el entorno digital.

Currently, traditional technologies have fallen behind in providing adequate protection against cyberattacks. Although antivirus software can identify and detect viruses by recognizing specific patterns, it has become insufficient against modern attacks that exploit cognitive biases and user vulnerabilities. In response to this need, a virtual assistant capable of providing cybersecurity education was developed. Its objective is to enable users to identify potential exposure to cyberattacks. To provide more effective guidance, a malicious domain detection model was created, achieving a correct classification rate of 70 %. Ultimately, it was demonstrated that using a virtual assistant and receiving notifications regarding the specific type of potential attack when accessing a malicious domain increases the probability of successful protection by 50 %. These results highlight the importance of adopting proactive and educational approaches to enhance user awareness and cybersecurity in the digital environment.

Un antivirus está diseñado específicamente para examinar y eliminar cualquier *software* malicioso detectado en un dispositivo electrónico [1]. Sin embargo, la mayoría de los antivirus actuales, aunque ofrecen una versión gratuita, no proporcionan todos los beneficios de la versión de pago; por ejemplo, ofrecen un escaneo, pero este se basa únicamente en firmas. En la actualidad, con los avances tecnológicos, depender exclusivamente del *hash* de un virus, que puede modificarse con cualquier cambio insignificante como un punto, ya resulta obsoleto para la vasta industria de *malware*.

En el ámbito de la ciberseguridad, la estrategia actual de los antivirus ya no es suficiente para educar a los usuarios y enseñarles a ser más precavidos. Con el crecimiento de la inteligencia artificial y la popularización de ChatGPT, el apoyo a los usuarios en diversas áreas se ha vuelto una realidad, y la educación en ciberseguridad no es una excepción. Es por ello que el asistente virtual propuesto a continuación busca centrarse en el usuario, brindándole educación mientras le ofrece opciones sobre cómo mantenerse seguro al navegar en la *web*.

Durante y después de los años de pandemia, no sorprende el aumento de los ciberataques, dada la creciente dependencia de la población en el uso de dispositivos tecnológicos. Según un informe de estadísticas de ciberseguridad, se registró un incremento del 300 % en los ataques cibernéticos desde el inicio del COVID-19. La mayoría de estos ataques han tenido lugar a través de dispositivos móviles, constituyendo un 70 % de los fraudes [2].

En Guatemala, desde el año 2021, ha habido un notable aumento en los ataques de *phishing* dirigidos a instituciones bancarias. Esto es preocupante debido a que, según datos de seguridad, el 50 % de las brechas de seguridad se atribuyen a *phishing* o a ingeniería social [2]. Un ejemplo destacado es el Banco Industrial, que, ante la creciente cantidad de ataques dirigidos a sus usuarios, emitió constantes advertencias para que no compartieran información bancaria a través de enlaces recibidos por correo electrónico. Además, implementaron una campaña informativa para concientizar a los usuarios sobre posibles ataques, ofreciendo en su sitio *web* información detallada sobre diversos tipos de ciberataques y consejos para protegerse. Los ataques llegaron al extremo de generar resultados fraudulentos al buscar Banco Industrial en navegadores *web*, direccionando a páginas maliciosas que buscaban obtener información confidencial de los usuarios.

En la actualidad, la tecnología evoluciona de manera constante y aporta mejoras significativas a la calidad de vida humana. No obstante, estos avances conllevan riesgos que los usuarios suelen pasar por alto. Es por ello que se desarrollaron los antivirus como una solución ante los peligros en línea.

A pesar de que los antivirus ofrecen una solución para prevenir que los usuarios sean víctimas de ataques, la mayoría de ellos tiende a bloquear la instalación o el acceso a un sitio *web* por completo. Esto a menudo lleva a que los usuarios desactiven o incluso desinstalen la aplicación. En general, esta suele ser una solución correctiva, ya que una vez que el usuario decide instalar una aplicación o explorar un sitio sospechoso, el antivirus bloquea la acción sin proporcionar un acompañamiento informativo. Simplemente informa al usuario que se ha bloqueado una amenaza, sin ofrecer orientación sobre los riesgos a los que estuvo expuesto y cómo podría haberlos prevenido.

Como respuesta a esta necesidad, se pretende desarrollar un asistente virtual diseñado para

ofrecer un acompañamiento en ciberseguridad al usuario. Este asistente tendrá la función de proporcionar orientación sobre cómo mantenerse seguro y protegerse ante la amplia variedad de ataques cibernéticos mientras navega en línea.

3.1. Objetivo general

Evaluar el incremento de la probabilidad de que un usuario se proteja ante un ataque cibernético al proporcionarle detalles sobre el tipo de *malware* utilizado por un dominio malicioso.

3.2. Objetivos específicos

- Diseñar un asistente de seguridad basado en *large language models* que permita educar a los usuarios acerca de los diversos tipos de ataques cibernéticos y las medidas de protección recomendadas cuando se encuentran expuestos a una amenaza.
- Especializar un modelo de *large language models* en ciberseguridad mediante *fine tuning*.
- Desarrollar un modelo de detección de dominios maliciosos mediante redes neuronales y características léxicas y estadísticas de los dominios.

El proyecto tiene como objetivo principal crear una herramienta tecnológica innovadora que proporcione asistencia virtual a los usuarios durante sus interacciones en la *web*, enfocándose específicamente en la navegación segura. El enfoque se centrará en educar a los usuarios sobre definiciones de tipos de ataque y recomendaciones de protección, destacando prácticas seguras y alertando sobre posibles riesgos de seguridad.

El alcance del proyecto incluirá el desarrollo de un modelo de clasificación de dominios capaz de analizar la seguridad de los sitios *web*, identificando posibles amenazas como *malware*, *phishing* o *ransomware*. Es importante destacar que, aunque el asistente virtual brindará orientación sobre la seguridad en general, no se abordará la creación de un modelo avanzado para la detección de aplicaciones maliciosas. La implementación de esta funcionalidad quedará fuera del alcance de este proyecto, centrándose en cambio en otras medidas y consejos para prevenir a los usuarios sobre posibles ataques durante la navegación en línea.

Es importante destacar que el alcance del proyecto se limita a la creación del asistente virtual y su integración en navegadores *web* o dispositivos específicos, excluyendo el desarrollo de una aplicación independiente. A pesar de esta limitación, se buscará garantizar la compatibilidad y accesibilidad del asistente en una variedad de entornos tecnológicos para maximizar su utilidad.

5.1. Asistente virtual

Un asistente virtual (VA, por sus siglas en inglés) es una herramienta de *software* capaz de comprender las necesidades de sus usuarios (expresadas mediante texto o voz) y, como resultado, generar una respuesta en lenguaje natural. Los términos *chatbot* y asistente virtual suelen usarse de manera intercambiable, y es que un *chatbot* es una herramienta que reconoce patrones en las consultas de sus usuarios para generar una respuesta. Por lo que los *chatbots* llegan a considerarse asistentes virtuales en el momento en que utilizan inteligencia artificial, tanto para reconocer las necesidades de sus usuarios como para generar la respuesta [3].

Los inicios de los *chatbots* radican alrededor de 1960, cuando el Instituto de Tecnología de Massachusetts desarrolló a Eliza, el primer *chatbot* mediante técnicas de *natural language processing* con el objeto de demostrar que “la relación humano-computador es esencialmente superficial” [4].

5.1.1. ¿Cómo funciona?

Un asistente virtual es un conjunto de módulos que interconectados permiten el reconocimiento, comprensión y generación de lenguaje natural. Según Pieraccini [5], un asistente virtual se puede representar mediante tres componentes principales:

- *Natural language understanding*: esto se refiere al proceso de transformar el *input* del usuario a un conjunto de símbolos comprensibles para las máquinas.
- *Dialog manager*: es el módulo encargado de predecir los siguientes símbolos del texto de entrada. Básicamente es el que genera la respuesta a la consulta del usuario.
- *Natural language generation*: este es el último módulo de un asistente virtual, ya que, una vez se recibe la respuesta del *dialog manager*, este módulo es el encargado de la “traducción” de lenguaje de computadora a texto comprensible para los humanos.

5.1.2. Tipos

Si bien el uso más común de los asistentes virtuales es el servicio al cliente en cualquier página de *e-commerce*, en la actualidad, debido al significativo incremento de la popularidad de la inteligencia artificial, esta se ha expandido para lograr satisfacer una serie de necesidades en varias industrias. Dentro de los servicios más comunes brindados por un asistente virtual, según Hope [6], se encuentran:

- Servicio al cliente: en general, este tipo de asistente virtual se encarga de responder dudas y resolver inconvenientes con los clientes.
- Administrativo: estos asistentes tienen el rol de una secretaria, ya que se encargan del trabajo administrativo, como agendar reuniones, planificación de tareas, preparación de reportes, entre otras tareas administrativas.
- *Bookkeeping*: estos se encargan de llevar registros, principalmente económicos, de una empresa. Su función principal recae en el registro de finanzas, buscando oportunidades de ahorro y un mejor manejo de costos.
- *Data entry*: su función es la ejecución de una serie de tareas con un *input* de datos, tal como llenado de una base de datos, transcripción de un video o grabar notas sobre una reunión.
- *E-commerce*: no se debe confundir el servicio al cliente con aquellos asistentes orientados a *e-commerce*. En el caso de los de *e-commerce*, estos buscan registrar todos aquellos artículos que se han vendido o guardado en favoritos. Esta es una estrategia de *marketing* que básicamente luego permite recordar a los clientes sobre aquellos productos que ha guardado que aún le pueden interesar.
- *Marketing* en redes sociales: se encargan de la administración de una cuenta de red social. Esto implica la elaboración de *posts*, administración de tráfico, análisis de ventas y otros.
- Diseño gráfico: se encargan de tareas como el diseño de logos y páginas *web*. Están orientados al diseño y la creatividad ya que son capaces de generar *posters* para publicaciones y/o generar cualquier tipo de material visual.
- Tecnología de la información: tal como su nombre lo indica, se encargan de realizar tareas relacionadas con computación, tal como investigación en internet o el envío automático de correos. Dependiendo de sus habilidades, en algunas ocasiones hasta son capaces de identificar errores de *hardware*.

5.1.3. Relevancia en la educación

Desde la aparición de Siri en 2014, la popularidad de los asistentes virtuales ha incrementado notablemente hasta la popularidad actual de ChatGPT (el cual es un modelo de inteligencia artificial capaz de generar texto basado en el contexto y mensajes pasados) [7]. Desde sus inicios, los VA han sido utilizados en campos como la medicina, *marketing*, *e-commerce* (servicio al cliente), entretenimiento y sobre todo educación. En los últimos años, especialmente con la pandemia, muchos iniciaron con el uso de herramientas de inteligencia artificial que permitieran cerrar la brecha de información y conocimiento [8].

En el campo de la educación se han identificado dos tipos de VA. El primero siendo orientado a servicios, es decir, aquel que puede dar apoyo a los nuevos estudiantes en preguntas administrativas y servicios del centro educativo. Y el segundo orientado a enseñar, ya que actúa como un catedrático que provee nuevos conocimientos reforzando la interacción de profesor-estudiante [3].

La interacción profesor-estudiante se genera a través de las consultas de los estudiantes. Esta es una parte esencial en la mejora del rendimiento académico. Constantemente, una de las barreras en el aprendizaje es el hecho que los estudiantes dudan en preguntar sobre temas que no comprenden, debido a que temen sobre posibles burlas de compañeros o una retroalimentación negativa por parte del profesor. Es aquí donde emerge la relevancia de los asistentes virtuales, mejorando el aprendizaje de los estudiantes al responder inquietudes en cualquier momento del día; a su vez, permiten actuar como intermediario entre alumno y profesor dando paso a un aprendizaje autodidacta [3].

5.2. *Natural language processing*

Natural language processing es la tecnología base en el desarrollo de los asistentes virtuales. Durante años se ha buscado el poder implementar en los ordenadores la capacidad de comprender emociones y entender el lenguaje natural. Se encarga básicamente del estudio del lenguaje, la comunicación y el reconocimiento de texto [9].

5.2.1. *Natural language understanding*

Natural language understanding (NLU) es una subárea de *natural language processing* encargada de reconocer la intención del mensaje del usuario mediante la interpretación semántica y la representación sintáctica; básicamente, NLU permite mantener una conversación con un dispositivo electrónico. Para esto utilizan dos conceptos principales: intención y entidades [10]. Una intención (*intent*) hace referencia a una correlación entre lo que el usuario busca decir y la respuesta que debe dar el sistema [11]. Por su parte, una entidad es básicamente una categoría con la cual se busca la identificación y extracción de identidades en el texto ingresado por el usuario, esto con el objetivo de que una computadora pueda comprender lo que el usuario desea conocer [10].

La detección de intención se ha tomado como un problema de clasificación, en el cual a un modelo de *machine learning* se ingresa un texto y, mediante la vectorización del mismo y otras técnicas de procesamiento, es posible predecir la intención con base en una serie de clases ya definidas. En los últimos años, una de las estrategias que está siendo implementada es el uso de *self-attention networks* para la detección de la intención del usuario [12]. Estas redes eliminan la parte recurrente y, en su lugar, ponen atención “especial” a sí mismas para identificar dependencias en el texto y reconocer a qué palabras o frases debe prestarse más “atención”. Esto permite analizar una gran cantidad de información, siendo así más rápido y eficiente [13].

5.2.2. *Dialog management*

Esta es una de las partes esenciales para garantizar la experiencia del usuario al momento de utilizar el asistente virtual, ya que contiene un módulo de manejo de errores que se encarga de manejar la ambigüedad y la incertidumbre de la información que recibe. La función principal del

dialog manager es controlar y actualizar el contexto de las conversaciones [14]. Los contextos son básicamente frases que almacenan el objeto al cual el usuario se está refiriendo [11] y la estrategia de interacción define quién controla la conversación. Dentro de estas estrategias se definen tres principales, según Huang [14]:

- Dirigida por el usuario: en esta, es el usuario quien hace las preguntas y el sistema únicamente se encarga de responder.
- Dirigida por el sistema: ocurre lo opuesto a la estrategia anterior, es el sistema quien toma la iniciativa y el usuario únicamente responde las preguntas.
- Iniciativa mixta: en este caso, tanto el sistema como el usuario pueden tomar el control.

Para el desarrollo del *dialog management* se tienen dos estrategias principales, según Greyling [15]:

- *Handcrafted*: se basa en respuestas predeterminadas, mediante reglas previamente definidas. Si bien se considera una estrategia bastante resiliente con respecto a fallos y errores, tiene como desventaja que las respuestas suelen estar prediseñadas; por lo tanto, suelen ser más robóticas que humanas.
- Probabilístico o estocástico: en este caso, en lugar de establecer la estrategia de diálogo mediante una serie de reglas previamente definidas, este toma un enfoque “personalizado” al aprender las reglas de conversaciones reales.

5.2.3. *Natural language generation*

Este módulo es el encargado de generar una respuesta al usuario. Para esto existe una serie de estrategias, como la generación de respuestas mediante una plantilla, en la cual solo se agregan las respuestas correspondientes a cada usuario; este método se conoce como *gap-filling*. Posteriormente se desarrollaron respuestas de manera dinámica, esto implica que el sistema puede responder a escenarios imprevistos sin necesidad de que exista programación “defensiva” [16].

La generación de lenguaje natural dinámico originalmente se desarrolló mediante cadenas de Markov, en donde se predecía la siguiente palabra utilizando la actual y la relación entre cada palabra única. Algo interesante es que este modelo fue bastante común en la sugerencia de palabras utilizada en el teclado de los teléfonos inteligentes [16].

5.3. *Large language models (LLM)*

Son modelos de *machine learning* encargados específicamente de generar mensajes en lenguaje natural. Son modelos que aprenden a predecir la probabilidad de la siguiente palabra basados en los textos previos. Este tipo de modelos son bastante efectivos en temas generales, debido a que suelen estar entrenados con una gran cantidad de información y son capaces de generar oraciones coherentes y contextuales.

Dentro de los *large language models* más populares se encuentran GPT-3 y GPT-4 de OpenAI. Estos modelos cuentan con una arquitectura *transformer*. Con esta arquitectura se utiliza el mecanismo de *self-attention*, el cual permite medir la importancia de las diferentes palabras en el *input* ingresado por el usuario y, con base en esto, generar una respuesta.

5.3.1. Entrenamiento de los LLM

Los modelos LLM son entrenados bajo un método llamado aprendizaje no supervisado. Es decir, a estos modelos se les provee un extracto de texto y aprenden a predecir la siguiente palabra basándose en las anteriores. En estos casos, el modelo no cuenta con una etiqueta específica, sino que, en cambio, aprende las propiedades estadísticas del lenguaje mediante los datos de entrenamiento [17].

El proceso de entrenamiento consta básicamente de “alimentar” al modelo con una secuencia de palabras y con la solicitud de que sea capaz de predecir la siguiente palabra. Las predicciones del modelo luego son comparadas con la palabra siguiente real y los parámetros del modelo son actualizados para reducir la diferencia entre la palabra predicha y la palabra real. Este proceso se repite millones de veces para lograr que el modelo realice mejores predicciones a medida que utiliza más datos [18].

5.4. Inteligencia artificial

A través de los años, la tecnología ha buscado replicar cada una de las habilidades humanas con el fin de poder optimizar cada una de las tareas desarrolladas por humanos. La inteligencia artificial (IA) es un estudio sobre cómo construir programas computacionales que permitan a las computadoras desarrollar la inteligencia humana [19]. El objetivo principal de la IA es lograr simular ciertos aspectos de la inteligencia humana en un programa computacional [20]. En la actualidad, un claro ejemplo de esto es ChatGPT, el cual no solo es capaz de generar texto en lenguaje natural, sino a su vez puede resumir y comprender texto de entrada.

La idea de la inteligencia artificial nace del hecho de que las computadoras deberían de ser capaces de replicar las habilidades humanas, tales como: aprender, hablar y resolver problemas. Lo que se busca es que un programa sea capaz de simular habilidades cognitivas humanas, convirtiéndose en un “humano híbrido”.

5.4.1. Tipos de inteligencia

La inteligencia se define como la habilidad de aprender, entender y generar juicios u opiniones que son basadas en el razonamiento [21]. No es un aspecto de un solo factor, sino más bien está compuesto por el entorno, la educación y las experiencias. Debido a esto, nace la teoría de múltiples inteligencias, la cual afirma que existen nueve inteligencias humanas. Según Shaari y Matore [22], estas son:

- **Inteligencia verbal lingüística:** es la habilidad de comunicarse con otros, persuadirlos y recordar información.

- Inteligencia lógica matemática: se refiere a la habilidad del uso efectivo de los números, a la comprensión de los principios de causa y efecto y la habilidad de entender el conjunto de números básicos.
- Visual espacial: habilidad de detectar y visualizar figuras, espacios, colores y líneas.
- Inteligencia cinestésica: contempla las habilidades físicas como la coordinación, elasticidad, flexibilidad y balance.
- Inteligencia musical: es la habilidad de detectar ritmos, canciones y detectar o cambiar ritmos y tempos.
- Inteligencia interpersonal: implica la comprensión de los sentimientos, motivaciones, hábitos o deseos de quienes nos rodean; básicamente es la habilidad de actuar con responsabilidad afectiva con quienes socializamos.
- Inteligencia intrapersonal: a diferencia de la interpersonal, esta se refiere a la comprensión interna de sentimientos, deseos, fortalezas y debilidades. Básicamente es la habilidad del autoconocimiento.

Al igual que los humanos, la inteligencia artificial no cuenta con un único tipo de inteligencia, sino que a lo largo de los años, y con los beneficios de los avances tecnológicos, se han desarrollado cuatro tipos principales, según lo descrito por Huang y Rust [23]:

- Inteligencia mecánica: es la habilidad de realizar tareas o rutinas de manera automática sin la necesidad de recibir instrucciones directas o apoyo de una persona. Un ejemplo de esto son los robots, ya que estos están orientados a la realización de tareas físicas de manera automática sin la necesidad de una instrucción directa. Este tipo de inteligencia está basada en reglas, su conocimiento es a priori y tienen una percepción constante del entorno para reaccionar a algún tipo de cambio. Debido a que no pueden comprender su entorno, no son capaces de adaptarse a este y su conocimiento se amplía de manera irregular debido a la repetitividad de sus tareas.
- Inteligencia analítica: este tipo de inteligencia se puede considerar más compleja que la anterior debido a que, en lugar de realizar una tarea repetitiva, esta lo que busca es la habilidad de procesar información para la resolución de problemas. Sus habilidades se basan en el procesamiento de información, razonamiento lógico y destrezas matemáticas. Este tipo de inteligencia está presente en el análisis de datos y el aprendizaje de máquina (*machine learning*). Básicamente, el aprendizaje en este caso es a través del entrenamiento de largos conjuntos de datos, en los cuales se obtienen patrones diferenciables de los mismos.
- Inteligencia intuitiva: desarrolla la creatividad como una habilidad esencial para la resolución de problemas. Su característica diferencial con respecto a la inteligencia analítica es la comprensión, ya que este tipo de inteligencia busca ser lo más flexible, ajustándose a diversas situaciones (siendo lo más parecido a un humano). En este caso, es muy poco común que se cometa un error dos veces, debido a que la inteligencia intuitiva aprende de las experiencias.
- Inteligencia empática: la habilidad destacable de este tipo de inteligencia es la capacidad de reconocer y comprender emociones humanas. Una máquina se comporta como si fuera capaz de experimentar emociones humanas. La característica principal de este tipo de inteligencia es la experiencia, pero a diferencia de la inteligencia intuitiva, aquí se refiere a la destreza

de experimentar algo. Esto es uno de los grandes retos actuales de la computación, debido a que no se ha definido “cómo es que un sistema físico puede tener experiencias vívidas con cualidades aparentemente intrínsecas, como el color de un tomate o el picante de un taco”.

5.4.2. Habilidades de IA

Tal como existe un conjunto de inteligencias humanas, los sistemas de inteligencia artificial buscan tener inteligencia multidimensional. La relación de estas inteligencias permite el desarrollo de un conjunto de inteligencias combinadas que forman parte de una tipología de inteligencia artificial, la cual resume las cinco habilidades generales encontradas en los sistemas de inteligencia artificial del mercado actual, según Pantano y Scarpi [24]:

- **Lógica matemática:** se basa en la habilidad de las máquinas para ser capaces de resolver problemas analíticos complejos que requieren de un pensamiento lógico. Básicamente, analiza problemas y situaciones de forma lógica y encuentra soluciones acorde.
- **Social:** esto está relacionado con la habilidad de comprender las emociones humanas e interactuar con humanos; básicamente es un proceso de inteligencia interpersonal.
- **Visual - espacial:** percepción espacial con la habilidad de poder manipular objetos en el espacio, es la habilidad del sistema para comprender el espacio.
- **Verbal - lingüística:** en esto lo que se busca es la simulación del uso del lenguaje natural. Es el proceso mediante el cual una máquina se comunica con humanos. Esto comúnmente se ve en sistemas como *chatbots* o asistentes virtuales como Alexa, Siri, entre otros.
- **Velocidad de procesamiento:** es básicamente la puesta en práctica de la inteligencia mecánica. En este caso se toma en consideración la habilidad de realizar tareas repetitivas de manera rápida y eficiente. Por lo tal, esto involucra la velocidad para desarrollar tareas simples de manera repetitiva, que no estén relacionadas con problemas matemáticos, con comparaciones visual-espaciales o generación de una conversación fluida.

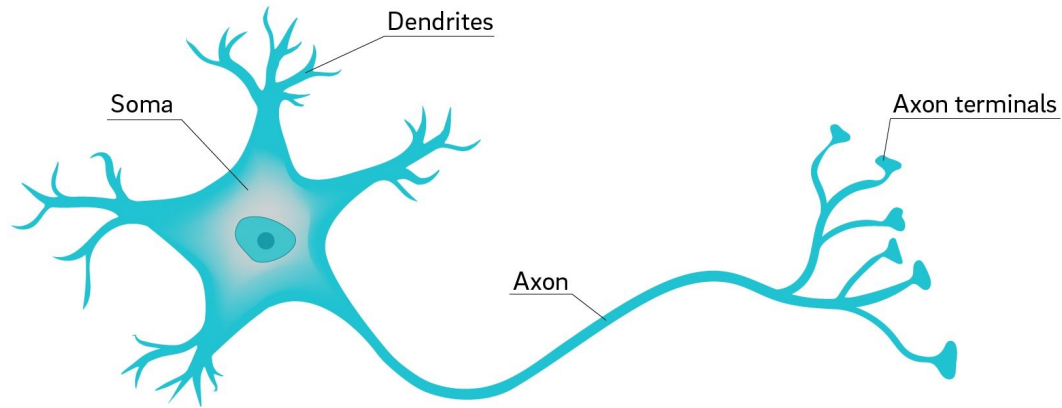
5.5. Redes neuronales

Las redes neuronales buscan imitar el funcionamiento básico del cerebro humano, especialmente el proceso de comunicación e interconexión de neuronas, con el cual se permita a sistemas computacionales la capacidad de realizar tareas complejas como el aprendizaje y la optimización [25]. Las redes neuronales son un algoritmo compuesto por unidades de procesamiento interconectado, organizado en capas [26]. Utilizan diversas aplicaciones de aprendizaje automático e inteligencia artificial para aprender, reconocer y generalizar patrones en los datos [27].

5.5.1. ¿Cómo funciona?

El cerebro humano consta de alrededor de 10 millones de células interconectadas entre sí. Cada una de estas células se conoce como neurona. Las neuronas se consideran como la unidad básica

Figura 1. *Componentes básicos de una neurona biológica*



Nota. Obtenido de [31]

de conocimiento, y son los elementos claves en el aprendizaje [28]. Las neuronas utilizan señales eléctricas y químicas para enviar información a distintos lugares del cerebro. Según el Instituto Nacional de Trastornos Neurológicos y Accidentes Cerebrovasculares [29], se componen de 3 partes básicas:

- Soma: también conocido como cuerpo celular. Este básicamente contiene el material genético y controla las actividades de la célula.
- Dendritas: es la encargada de recibir información para la neurona.
- Axón: es el encargado de enviar “mensajes” desde la neurona.

La parte más interesante de las neuronas no es solo su funcionalidad y su interconexión, sino el proceso de envío de señales a lo largo de los axones y su respectiva recepción mediante las dendritas. Como bien se describió, cada neurona recibe una serie de señales de otras neuronas a través de las dendritas. Si la suma de estas señales es lo suficientemente “fuerte”, activa la transmisión de señal a través del axón. Este proceso es iterativo, es decir, la señal se comparte a todas las dendritas conectadas al axón que dispara la señal, y si el total de la señal que recibe el soma excede un cierto nivel, entonces se envía nuevamente esta señal [30].

Este proceso es el que imitan las redes neuronales artificiales mediante entradas (dendritas), salidas (axones) y funciones de activación (señales enviadas a las neuronas). Las redes neuronales

se conforman de capas, que a su vez, están compuestas por neuronas, las cuales se interconectan mediante funciones de activación, que básicamente condicionan el proceso de propagación de un valor (señal).

En la actualidad, muchas de las redes neuronales desarrolladas están compuestas en capas de nodos y se “alimentan hacia adelante”, es decir, un nodo (neurona) en particular está conectado a varios nodos en la capa anterior, de los cuales recibe información, y a su vez, está conectado a varios nodos en la capa posterior a los cuales les envía información. A la información recibida en los nodos, se le asigna un peso. Cuando la neurona se “activa” recibe un valor diferente en cada una de sus conexiones, el cual es multiplicado por el peso asociado a la conexión. Luego, si la suma ponderada de estos pesos excede un valor de *threshold*, entonces el nodo envía el valor a través de todas sus conexiones disponibles. Por el contrario, en caso de que la suma ponderada no exceda el valor límite, el nodo no envía ningún valor [27].

5.5.2. Arquitecturas

El modelo más simple de redes neuronales se conoce como perceptrón. En 1957 fue la primera red neuronal entrenable. Consiste básicamente en una capa de entrada (con varias neuronas), que transmite la información directamente a la capa de salida, la cual tiene una sola neurona. Esta neurona procesa las señales recibidas, es decir, realiza la suma ponderada de los pesos y envía la información a la función de activación, permitiendo así generar una clasificación binaria [32]. Si bien, en su momento, la idea del perceptrón fue bastante innovadora, este no imitaba con exactitud la complejidad del funcionamiento del cerebro humano, además de que contaba con la limitación de que únicamente podía resolver problemas linealmente separables en dos categorías. Con base en esto, nacen una serie de arquitecturas de redes neuronales que permiten la resolución de problemas más complejos, como la clasificación de objetos en una imagen, la generación y comprensión de lenguaje natural e incluso la generación de imágenes.

5.5.3. Perceptrón multicapa

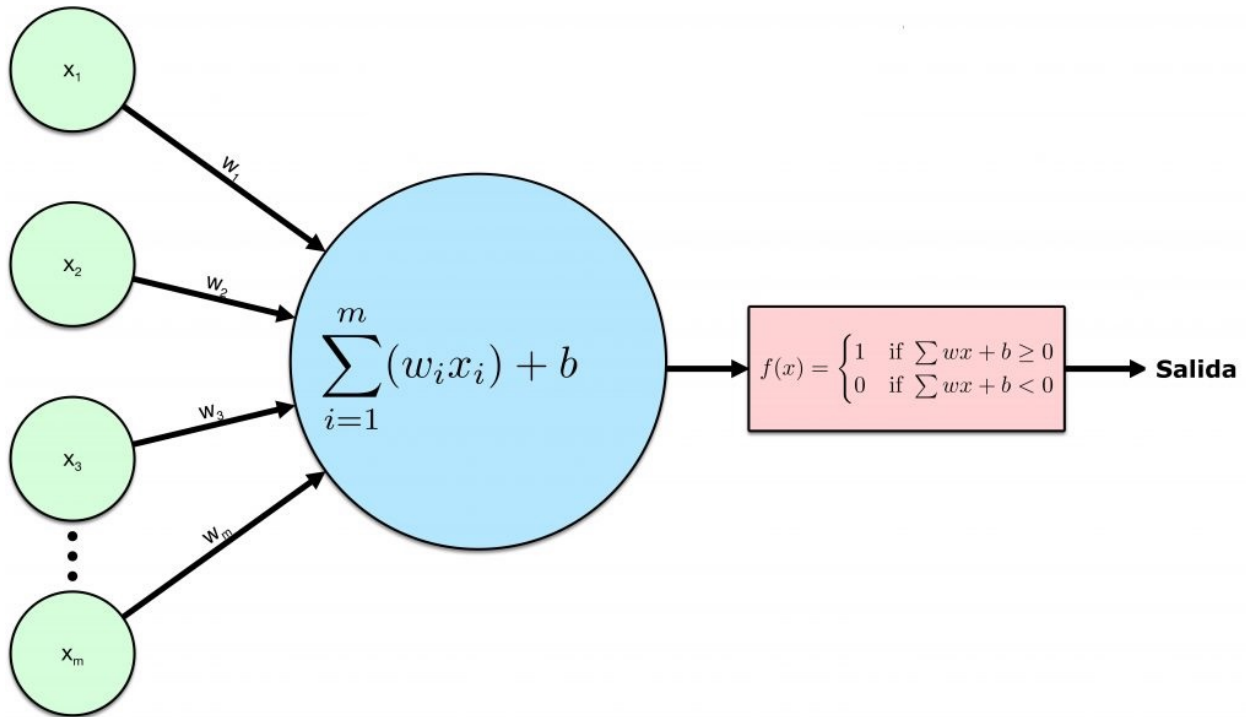
También conocidas como *feed-forward neural networks*, ya que en este caso la información viaja en una sola dirección, es decir, ingresan las variables de entrada, pasan por las capas “ocultas” y se envían a la capa de salida. No existe un proceso de retroalimentación en donde la salida del modelo sea nuevamente una entrada.

Este tipo de redes neuronales busca resolver las limitaciones de los perceptrones. En este caso se cuenta con una capa de entrada, con una o varias capas “ocultas” y una capa de salida. Las capas intermedias u ocultas adquieren su nombre debido a que el comportamiento o resultado de estas capas no está directamente descrito en los datos de entrenamiento, a diferencia de la capa de salida en la cual el resultado debe de ser el valor o *label* asignado a los datos de entrenamiento [34].

5.5.4. Redes neuronales recurrentes (RNN)

Estas redes neuronales son utilizadas para trabajar con datos secuenciales (como cadenas de texto). Utilizan la idea de compartir parámetros en diferentes partes del proceso de entrenamiento

Figura 2. *Arquitectura de un perceptrón*



Nota. Obtenido de [33]

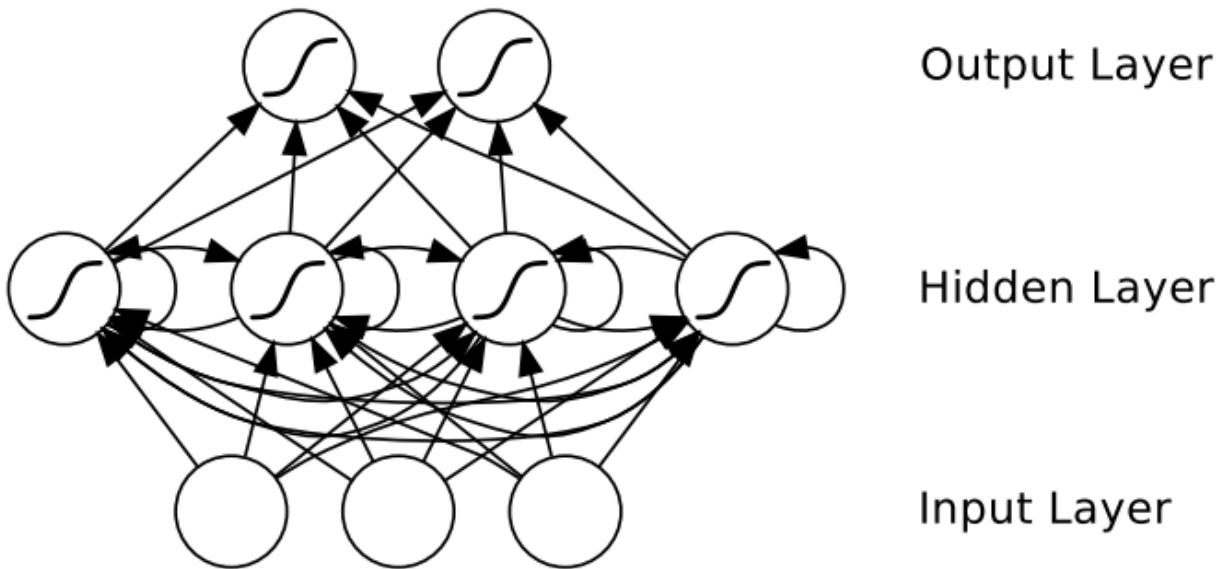
del modelo [34]. A diferencia de las redes multicapa, en la cual la información viaja en una única dirección, en este caso se admite que las capas internas tengan retroalimentación de la salida de otras capas, tal como se muestra en la Figura 3. Básicamente, lo que ocurre en este caso es que las capas comparten como entrada las salidas de otras capas, permitiendo así que exista una especie de memoria de las entradas anteriores que persiste en la red y, por ende, influencia el *output* generado [35].

5.5.5. *Generative adversarial nets (GAN)*

Consiste en dos redes neuronales, un generador y un discriminador, que son entrenados de manera adversaria; es decir, mientras que el generador toma datos aleatorios de entrada y genera datos sintéticos que asegura ser indistinguibles de los datos reales, el discriminador debe estimar la probabilidad de que una entrada sean datos que pertenecen al conjunto de entrenamiento o datos generados por el generador. Su uso radica principalmente en la generación de datos falsos, como imágenes, audio o texto mediante el aprendizaje de la distribución de datos reales [26].

En analogía, el modelo generativo se puede comprender como un falsificador, el cual busca producir piezas de arte falsas y venderlas sin ser detectado. Por su parte, el modelo discriminador se puede pensar como la policía que intenta detectar las falsificaciones. La ventaja del entrenamiento adversario es el hecho de que ambos modelos buscan mejorar sus habilidades hasta el punto de que

Figura 3. Arquitectura base de una red neuronal recurrente



Nota. Obtenido de [35]

los datos falsificados sean indistinguibles de los genuinos [26].

5.5.6. *Transformer*

Esta arquitectura de red neuronal utiliza el mecanismo conocido como *self-attention*, el cual es una función que se describe como un mapeo entre un *query* y un conjunto de claves-valores a un valor de salida. Mediante esta función el valor de salida se genera como la suma de los pesos de los valores; estos pesos, a su vez, son generados a través de una función de compatibilidad entre el *query* y la clave. Algo interesante a destacar es el hecho de que *self-attention* busca relacionar diferentes posiciones en una secuencia; esto permite que sea capaz de capturar la relación y dependencia entre los elementos de dicha secuencia, permitiendo así poder computar una representación de la secuencia [17].

Los principales componentes en una arquitectura *transformer*, según Vaswani et al. [17], se describen a continuación:

- *Multi-head self-attention*: esto permite al modelo ponderar la importancia de diferentes palabras en una oración al momento de hacer predicciones, siendo adecuado para capturar dependencias a largo plazo en las secuencias.
- *Positional encoding*: debido a que los modelos de tipo *transformer* carecen de información

secuencial de manera inherente, es necesario codificar las posiciones de las entradas para proporcionar al modelo información sobre las posiciones de una palabra en una secuencia.

- *Feed-forward networks*: luego de las capas de atención, el modelo utiliza redes neuronales multicapa para procesar las características de la información.

5.5.7. *Generative pre-trained transformer (GPT)*

Generative pre-trained transformer es un modelo desarrollado por OpenAI. Su funcionalidad radica en la generación de lenguaje natural, basado en una entrada ingresada por el usuario. Para esto utiliza como base la arquitectura *transformer* y la entrena sobre un gran corpus de datos. Se dice que es un modelo preentrenado debido a que inicialmente el modelo se entrena sobre una gran cantidad de información de internet, la cual le permite aprender gramática, semántica y conocimiento en general. Posterior a esto, el modelo es afinado para aprender tareas del lenguaje específicas como traducción, completación de texto o resumen de información [36].

El modelo generativo preentrenado presenta cuatro características clave:

- Generación autorregresiva: el modelo de predicción del modelo generativo se basa en predecir una palabra a la vez, basado en las palabras anteriormente generadas [37].
- Arquitectura a gran escala: cuentan con una gran cantidad de parámetros y *tokens* de entrenamiento que contribuyen a su alto desempeño [38].
- Entendimiento del contexto: debido a sus dos fases de entrenamiento, poseen una gran capacidad para comprender el contexto del lenguaje, lo que les permite la generación de texto coherente y contextualmente apropiado [36].
- Versatilidad: debido a que son entrenados bajo aprendizaje no supervisado, estos luego pueden ser “afinados” (*fine-tuned*) para una tarea en particular de lenguaje natural [36].

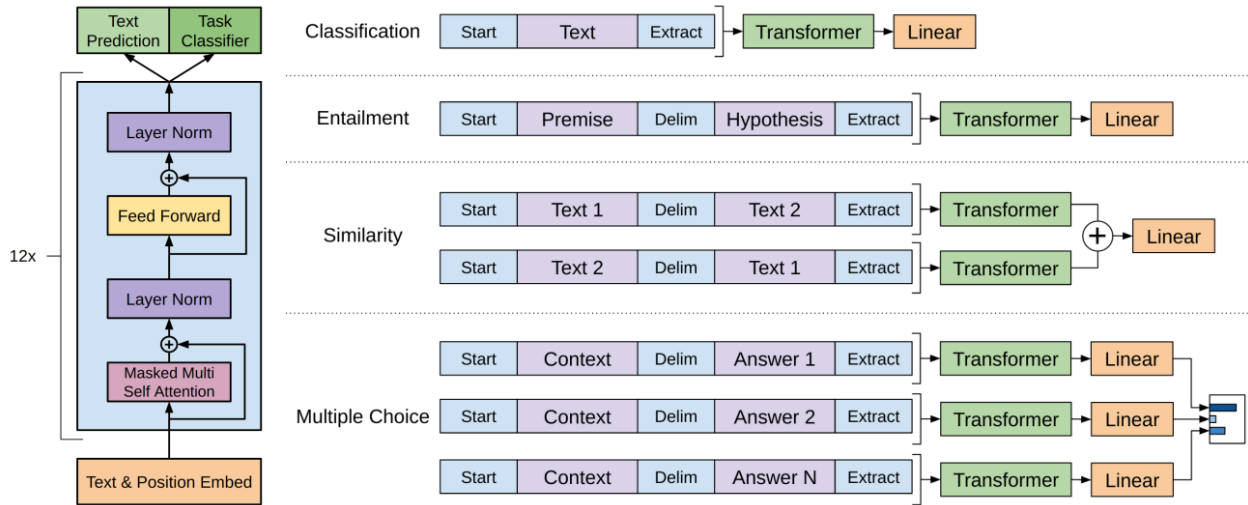
¿Cómo funciona?

El funcionamiento de GPT se basa principalmente en el uso de *transformers* como arquitectura de red neuronal y de dos etapas de entrenamiento, que permiten el aprendizaje generalizado y la especialización del modelo [38].

- Entrenamiento no supervisado: en esta etapa se busca que el modelo aprenda de manera generalizada el lenguaje natural. Para ello se utiliza una red multicapa *transformer decoder*, la cual se preentrena sobre una gran cantidad de datos con los cuales se busca que el modelo aprenda patrones estadísticos mediante el retorno de las distribuciones de los *tokens* de entrada.
- *Supervised fine-tuning*: en este caso se tiene un conjunto de datos etiquetados, donde cada instancia corresponde a una secuencia de *tokens* de entrada junto con una etiqueta. Este proceso permite la adaptación del modelo para tareas particulares como resumen, traducción o *question answering*.

- Generación de texto: como bien se mencionó, para la generación de texto utiliza un sistema autorregresivo, utilizando la palabra que acaba de predecir para generar la siguiente palabra. Para seleccionar la siguiente palabra a generar, el modelo puede utilizar tres estrategias: *greedy sampling*, escoge la palabra que tiene la mayor probabilidad; *random sampling*, toma una muestra aleatoria basado en las probabilidades predichas; y finalmente, *beam search*, mantiene una serie de secuencias probables y selecciona la de más alta probabilidad.

Figura 4. Arquitectura simple de GPT



Nota. Obtenido de [36]

5.6. Seguridad de la información

Seguridad se refiere a la protección de algo o alguien contra amenazas o posibles ataques [39]. Por su parte, la información se define como conocimiento, hechos y/o noticias [40]. Es común pensar que seguridad de la información es la protección de la información en dispositivos tecnológicos; sin embargo, al ser un concepto multidimensional, hace referencia a la protección de los activos de la información (sistemas y *hardware* utilizados para almacenar o transmitir la información), ya sea en formato digital o físico [41].

La seguridad de la información busca asegurar la integridad, disponibilidad y confidencialidad de los datos evitando el acceso no autorizado, divulgación, alteración y/o destrucción. No solo busca resguardar sobre riesgos tecnológicos, sino a su vez factores humanos, desastres naturales y fallas técnicas [41].

5.6.1. Características de la información

Una de las métricas dentro de la seguridad de la información se conoce como la tríada de la información. Estas son las características fundamentales de la información que permiten asegurar los activos de la misma. El valor de la información se mide con base en las características que posee; básicamente, si una característica cambia, el valor de la información puede incrementar o disminuir. En la actualidad, la tríada ya no es suficiente para asegurar el valor que tiene la información para los usuarios, es por ello que se ha creado una tríada expandida con las siguientes características, según Whitman y Mattord [42]:

- **Confidencialidad:** asegura que la información es solamente accesible para los sistemas o usuarios autorizados, y protege contra la divulgación no autorizada. Generalmente, esto se logra a través de encriptación y controles de acceso.
- **Integridad:** la información es confiable, precisa e inalterable; busca que la información, como un todo, sea completa e incorrupta. Se encarga de proteger la información de modificaciones no autorizadas, ya sean accidentales o no.
- **Disponibilidad:** define que la información sea accesible para los usuarios o sistemas autorizados cuando lo necesitan. Esta característica es esencial, ya que asegura el valor de la información en el tiempo y puede ser crucial en la toma de decisiones.
- **Autenticidad:** se refiere a la cualidad de ser genuina, es decir, que la información no sufra ninguna transformación y esté en el mismo estado en el cual fue creada, almacenada o transferida. A su vez, asegura que la información es lo que asegura ser con fuentes verificables y creíbles.
- **Utilidad:** la información posee valor para algún propósito o usuario. Para esto, debe de cumplir con ser fácil de entender, acceder y utilizar.
- **Posesión:** esto se refiere a la cualidad de control o propiedad de la información. Conlleva la responsabilidad de asegurar la confidencialidad e integridad de la información.

5.6.2. Ciberseguridad

Se puede considerar como una subrama de la seguridad de la información, que se encarga específicamente de los activos de la información que poseen algún elemento digital. Se encarga de proteger computadoras, redes o conexiones y la información digital. En este caso, el riesgo principal son los ataques cibernéticos, como acceso no autorizado, *hacking*, *malware*, divulgación de información confidencial o alguna otra actividad maliciosa [43].

5.6.3. El factor humano de la ciberseguridad

Al igual que la seguridad de la información, la ciberseguridad es un concepto multidimensional. Sin embargo, las personas suelen centrarse en los aspectos tecnológicos, configuración de un buen *firewall*, encriptación de los datos o utilizar un antivirus. Esto ocurre principalmente debido a que centrarse en aquellos elementos que se pueden controlar conlleva a pensar que se está seguro. El problema con esto es que se deja de lado a quien interactúa con esos sistemas, al humano [44].

El factor humano en la ciberseguridad se refiere al papel que juegan las personas en la seguridad de los sistemas. Las acciones, el comportamiento y las decisiones pueden, ya sea mejorar o comprometer la ciberseguridad de un sistema; es por ello que el entrenamiento y la concientización son esenciales, ya que permiten educar al usuario sobre posibles riesgos en línea y/o prácticas seguras. Esto puede evitar que una persona sea víctima de algún ataque que termine comprometiendo la seguridad de todo el sistema [45].

En la actualidad, una de las formas más comunes por la cual los atacantes buscan explotar las vulnerabilidades humanas es mediante la ingeniería social. A grandes rasgos, “la ingeniería social es cualquier acto que influencia a una persona a tomar una acción que puede o no estar dentro de sus mejores intereses” [46]. Algo interesante de la definición anterior es el hecho de que la ingeniería social no necesariamente debe de ser algo negativo, sino más bien es una forma mediante la cual se puede llegar a influenciar a alguien. El problema surge cuando una persona entiende cómo es el proceso de toma de decisiones y utiliza *triggers* emocionales y principios psicológicos mediante ingeniería social para que una persona tome una acción que no lo beneficia, como la divulgación de información confidencial [46].

5.7. Psicología y ciberseguridad

La psicología es la ciencia que estudia el funcionamiento de la mente humana, así como su influencia en el comportamiento [47]. Involucra el estudio de procesos cognitivos, experiencias emocionales, interacciones sociales y la base biológica del comportamiento [48].

Como bien se mencionó, el factor humano es un aspecto clave en el proceso de ciberseguridad. El comprender cómo funciona la mente humana no solo permite a los atacantes explotar vulnerabilidades, como los sesgos cognitivos, sino también da paso al desarrollo de estrategias que permitan la mitigación de vulnerabilidades de seguridad a causa de factores humanos [49].

Mediante un estudio se demostró la relación que existe entre la liberación de oxitocina (la cual comúnmente se conoce como la hormona del amor, debido a su influencia en los lazos sociales) y la confianza que se cree que alguien más proporciona. Es decir, no solo está involucrada en el proceso de relaciones sociales, la creación de empatía y generar confianza con otros, sino que también cuando se cree que alguien más ha dado su confianza. Por su parte, también se tiene la dopamina, la cual es un neurotransmisor que se produce en momentos de placer, felicidad o procesos cognitivos como el aprendizaje o la atención. La combinación de estas dos, oxitocina y dopamina, genera lo que se conoce como un “cóctel hormonal de ingeniería social, el cual es capaz de abrir cualquier puerta” [46].

5.7.1. Sistemas de pensamiento

El pensamiento o cognición es un proceso mental que implica la manipulación de información como la adquisición, procesamiento, almacenamiento y memoria. Consta de tres fases principales: recepción, percepción y recuperación de información [50]. La información manipulada o adquirida mediante el pensamiento permite la resolución de problemas, toma de decisiones y generación de razonamiento [51].

La psicología cognitiva es la ciencia que se encarga de estudiar el proceso de cognición y, básicamente, cómo se realizan los procesos cognitivos. A lo largo de los años, una de las teorías más predominantes en psicología ha sido sobre la existencia de dos modelos de procesamiento de información utilizados en la generación de juicios, resolución de problemas y toma de decisiones. Esta teoría se conoce como *dual process theory* y básicamente indica que existen dos sistemas de pensamiento: el primero, que se caracteriza por ser automático y rápido; y el segundo, controlado y consciente [52].

En 2011, el psicólogo Daniel Kahneman profundizó no solo en la teoría del procesamiento dual, sino también en el efecto de las heurísticas y sesgos cognitivos como “atajos mentales” en la toma de decisiones, proponiendo dos sistemas descritos en [53]:

- Sistema 1: involucra procesos mentales subconscientes, rápidos y sin esfuerzo que permiten una rápida generación de juicios y toma de decisiones basada en heurísticas. Este sistema controla alrededor del 98 % del pensamiento y se constituye principalmente de habilidades innatas, como el reconocimiento de objetos, pero también de aquellas actividades que se practican de manera constante. Se caracteriza por ser rápido, automático e intuitivo.
- Sistema 2: es el responsable aproximadamente del 2 % del pensamiento e involucra procesos mentales que requieren atención, razonamiento y un compromiso cognitivo más profundo (en comparación con el sistema 1). A su vez, es el encargado de “controlar” el comportamiento, tal como ser amable aun cuando se está enojado; está “programado” a incrementar el nivel de esfuerzo cuando considera que un error está por cometerse. Se caracteriza por ser deliberado, consciente y analítico, y básicamente “aparece” cuando el sistema 1 es incapaz de responder a una problemática planteada.

Si bien pareciera ser que el sistema 2 es quien debería de “gobernar” el proceso de pensamiento, su uso continuo puede generar una sobrecarga de uso de procesos cognitivos, lo cual puede ocasionar fatiga mental. Es por ello que se considera que la relación existente entre ambos sistemas es altamente eficiente, ya que minimiza esfuerzos y optimiza el rendimiento [53].

El hecho de que el sistema 1 sea mucho más rápido que el sistema 2 se debe a que cuenta con heurísticas y sesgos que permiten que el proceso de pensamiento sea mucho más veloz. Estas heurísticas se forman a través de la conexión que se realiza mediante la información que se percibe y las experiencias que se han vivido; por lo tal, las heurísticas se forman a través de experiencias pasadas. El problema con estas heurísticas surge cuando, ante un problema, se centra la atención únicamente en un aspecto, ignorando el resto; este tipo de heurísticas se conocen como sesgos cognitivos [54].

5.7.2. Sesgos en ciberseguridad

La ciberseguridad está “formada” por un conjunto de decisiones tomadas por seres humanos. Y las decisiones son principalmente guiadas por el sistema 1 en conjunto con heurísticas y sesgos cognitivos [54]. Una de las principales razones por las cuales es importante considerar los sesgos es debido a que los atacantes “apuntan” a los sesgos cognitivos en combinación con fallas en el sistema para lograr una mejor explotación de dichas fallas [55].

Con esta combinación, el atacante tiene el ambiente perfecto para desarrollar su amenaza, con decisiones sesgadas (lo que puede conllevar a error humano) y error tecnológico. Existen un sinnúmero

de sesgos cognitivos, y varios de ellos se encuentran altamente relacionados con el manejo de la información en la red; sin embargo, el aspecto esencial en esto es el poder conocer cómo delimitar la influencia de dichos sesgos en la toma de decisiones.

Efecto avestruz

El desinterés de las personas por informarse sobre temas que consideran que no son su responsabilidad o no les parecen agradables forma parte de lo que se conoce como el efecto avestruz. Este nace de la falsa idea de que el avestruz esconde su cabeza bajo la arena para ignorar el peligro o pretender que este no existe [56].

Frecuentemente se escucha el típico “que la empresa se encargue de la seguridad de la información, a mí solo me contrataron por servicio al cliente, ¿Para qué me van a dar una capacitación al respecto?”. Cuando la información se percibe como negativa o desagradable, los usuarios son menos receptivos, lo que puede conllevar a no tomar en consideración las recomendaciones y buenas prácticas de seguridad, simplemente porque no se está dispuesto a hacer un esfuerzo extra [55].

Ilusión de invulnerabilidad

“Eso nunca va a pasarme a mí” o, el típico, “para qué quieren mi información, no tengo dinero y tampoco soy famoso”, son ejemplos claros del sentimiento de invulnerabilidad ante un ataque cibernético.

Es uno de los sesgos más comunes dentro de la ciberseguridad; de manera popular se conoce como el sesgo del optimismo y consiste en creer que es mucho más probable que suceda un evento positivo en contraste con uno negativo; básicamente se sobreestima la probabilidad de los eventos con resultado positivo. Esto va de la mano con la ilusión del control: como seres humanos, buscamos tener el control de las decisiones que tomamos, y el optimismo da la capacidad de percibir que se tiene control del entorno [57].

Algo interesante respecto a este sesgo es el hecho de que se modifican creencias o probabilidades de que ocurran ciertos eventos en la vida con mayor facilidad cuando se está expuesto a información positiva. Esto se demostró mediante un estudio en el cual se le preguntó a los participantes que indicaran la probabilidad de que les sucediera una serie de eventos (tanto positivos como negativos); luego se les indicó la probabilidad “real” de experimentar cada una de las situaciones, y se les pidió nuevamente que indicaran su probabilidad de experimentarlo. En los resultados se pudo observar que si la probabilidad que habían indicado los participantes inicialmente era mucho más baja que la real, esta era levemente modificada (en algunos casos permanecía igual). Por el contrario, si la probabilidad inicial dada por el participante era mucho más alta que la real, entonces esta disminuía considerablemente [57].

6.1. Datos

A lo largo del desarrollo del proyecto, se utilizaron dos conjuntos de datos esenciales para entrenar dos modelos de aprendizaje profundo (*deep learning*). El primer conjunto se empleó en el proceso de *fine tuning* del modelo GPT-3.5 de OpenAI. En cuanto al segundo conjunto de datos, se dedicó específicamente al entrenamiento del modelo de detección de dominios.

Los datos recopilados para el *fine tuning* se obtuvieron de Wikipedia, una enciclopedia libre que proporciona una amplia variedad de información sobre diversos temas. Además, se consultaron *blogs* informativos de reconocidos proveedores de antivirus en el mercado, como Kaspersky y Microsoft, para obtener detalles sobre cómo protegerse contra distintos tipos de ataques.

En relación con el modelo de detección de dominios, las categorías de dominios se obtuvieron de fuentes diversas. Los dominios benignos se adquirieron a través de Cisco Umbrella, una empresa de seguridad en la nube que ofrece un servicio de DNS con información sobre dominios. Por lo tanto, la empresa decidió crear una lista de un millón de dominios más populares, accesible de forma gratuita [\[58\]](#).

En cuanto a los dominios de *phishing*, *malware* y *ransomware*, se utilizaron conjuntos de datos proporcionados por dos repositorios que buscan mantener una lista actualizada de dominios maliciosos para ser utilizada como *blacklist* en navegadores o antivirus (ver Anexos).

6.2. Limitaciones

Entre las limitaciones preeminentes del modelo *fine-tuned* de GPT, se destaca su dificultad ocasional para generar textos breves y concisos al ser consultado sobre temas específicos. A pesar de

haber pasado por un proceso de entrenamiento, el modelo muestra una propensión a la alucinación, es decir, la tendencia a proporcionar información que no siempre es precisa. Además, de manera general, se reconoce que este modelo carece de la capacidad intrínseca para detectar de forma autónoma cualquier tipo de ataque cibernético.

Por otro lado, en cuanto a las limitaciones del modelo de clasificación de dominios, se enfrenta al desafío de clasificar dominios maliciosos que no sigan patrones generados mediante un algoritmo (DGA). Asimismo, debido a las similitudes entre los dominios de tipo *malware* y *phishing*, el modelo presenta dificultades para reconocer la diferencia entre ambos, lo que puede resultar en una clasificación errónea.

7.1. Selección de modelo de LLM

El proceso de selección del modelo a utilizar consistió en una búsqueda de distribuidores de modelos LLM, tales como OpenAI, Google o TII. Algunas de estas empresas cuentan con modelos *open source*, es decir, proporcionan directamente los modelos preentrenados con sus respectivos pesos. La implementación de los modelos *open source* puede realizarse mediante la descarga de una librería en Python, conocida como Autotransformers de *Hugging Face* o, en algunos casos, mediante el repositorio provisto por la empresa. Dentro de los modelos de generación de lenguaje natural *open source* más famosos se encuentran:

- BERT: este es un modelo desarrollado por Google, el cual busca comprender el contexto de las palabras en una oración, considerando tanto las palabras que están detrás como las que están delante [59].
- LLaMA: es un modelo desarrollado por Meta; la idea detrás de este modelo es poder proporcionar a todas aquellas personas que carecen de recursos de infraestructura un modelo que utilice una menor cantidad de recursos computacionales. Para esto proveen una serie de modelos entrenados en una diversa cantidad de *tokens*, desde el modelo 7B, entrenado en 7 mil millones de *tokens*, hasta el modelo 65B, entrenado en 65 mil millones de *tokens* [60].
- Falcon: fue desarrollado por el Instituto de Tecnología en Innovación. Uno de sus modelos (Falcon 40B) estuvo liderando un *ranking* de modelos *open source* en *Hugging Face*. Lo destacable de este tipo de modelo es el hecho de que su entrenamiento se basa en datos de “alta calidad”, ya que el equipo de desarrollo cuenta con un *pipeline* que les permite extraer el texto de mejor calidad al aplicar una gran cantidad de filtros y eliminar textos duplicados [61].

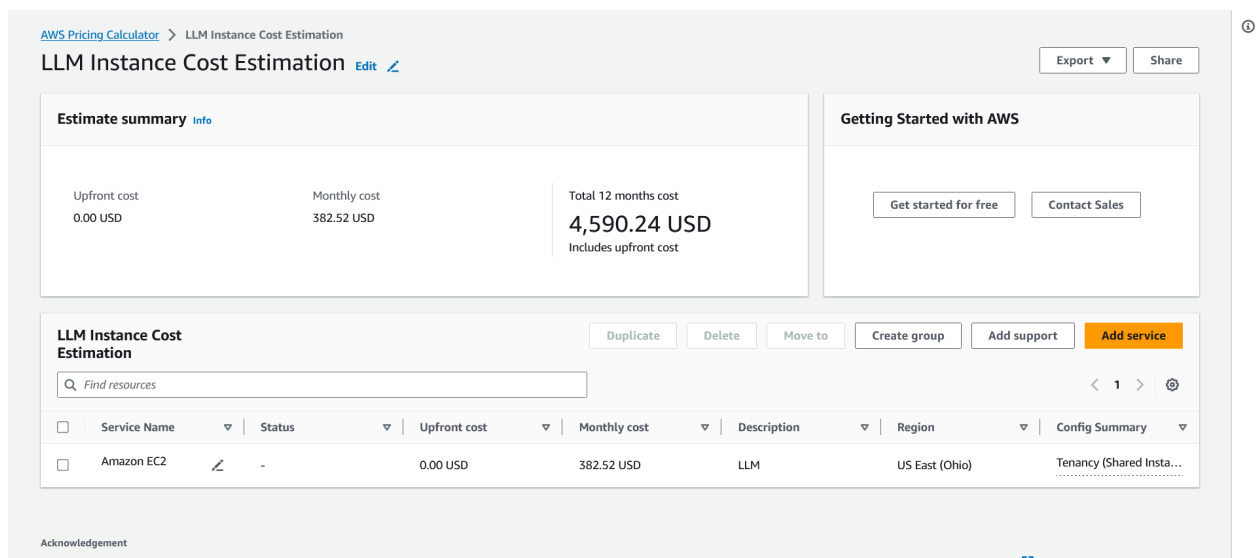
Si bien cada uno de estos modelos posee características sobresalientes en el mercado, además del

hecho de ser totalmente gratuitos, al momento de la selección del modelo se contempló el costo de implementación.

El uso final del modelo será en una aplicación móvil; por lo tanto, uno de los requisitos de todos los modelos es que deben ser rápidos y consumir pocos recursos. Esto no es algo viable si se utilizan de manera directa los modelos *open source* debido a que cuentan con una serie de “pesos” y configuraciones (del proceso de preentrenamiento) que, por ejemplo, para el modelo Falcon 40B utilizan alrededor de 45 GB de RAM para poder ser ejecutados. Esto no toma en consideración los recursos necesarios para poder hacer *fine tuning* del modelo.

Debido a que este tipo de modelos consumen demasiados recursos físicos, una alternativa que se estudió fue el hecho de utilizar una instancia la cual contara con los recursos suficientes para el modelo *open source*. Esto implicaría el desarrollo de una API que permitiera la conexión de la aplicación con el modelo. Sin embargo, ante esto hay que considerar el costo de una instancia en un proveedor de servicio en la nube. Para ello, se realizó una cotización en AWS con una instancia con los requisitos de *hardware* para ejecutar el modelo Falcon 40B, y como se puede observar en la figura 5, el costo mensual de la instancia era de aproximadamente \$390.00, lo cual representa un costo de Q3,042.00; esto hace que el costo de mantenimiento y de sostenibilidad de la aplicación para los meses de prueba sea inviable.

Figura 5. Cotización de uso de instancia en Amazon EC2



Por su parte, también se cuenta con empresas que proveen modelos pagados, como OpenAI, el cual cuenta con una API tanto para el consumo de los modelos como para el proceso de *fine tuning*. Para utilizar estas API es necesario tener una clave que ellos proveen al momento de crear una cuenta en el servicio. El sistema de cobro de OpenAI es con base en la cantidad de *tokens* que se envían en una respuesta o se consultan al modelo. Este modelo de costos, tal y como se puede observar en la figura 6, es por cada mil *tokens* de respuesta o de solicitud. Esto permite que el cobro sea bajo demanda, es decir, se va a tener un cobro únicamente cuando los usuarios consulten algo mediante la aplicación. Esto representa una mejora con respecto a la propuesta de una instancia debido a

que una instancia generalmente cobra por horas encendidas, se esté utilizando o no algún recurso en esta. De igual forma, el sistema de cobro para hacer *fine tuning* del modelo se basa en cada mil *tokens* enviados para entrenamiento y los precios exactos se pueden observar en la figura 7.

Figura 6. Costos por uso de modelo GPT-3.5 de OpenAI

Model	Input	Output
4K context	\$0.0015 / 1K tokens	\$0.002 / 1K tokens
16K context	\$0.003 / 1K tokens	\$0.004 / 1K tokens

Figura 7. Costo de proceso de *fine tuning* en modelos de OpenAI

Model	Training	Input usage	Output usage
babbage-002	\$0.0004 / 1K tokens	\$0.0016 / 1K tokens	\$0.0016 / 1K tokens
davinci-002	\$0.0060 / 1K tokens	\$0.0120 / 1K tokens	\$0.0120 / 1K tokens
GPT-3.5 Turbo	\$0.0080 / 1K tokens	\$0.0120 / 1K tokens	\$0.0160 / 1K tokens

Si se estima una cantidad máxima de 50 usuarios para el proceso de elaboración de este trabajo, con una duración de pruebas máxima de 30 días, se tiene que el costo aproximado de utilización del modelo sería alrededor de \$33.00, lo cual representa un 8% del costo total de utilizar una instancia con un modelo *open source*. Debido a esta relación de costo-beneficio, se optó por utilizar modelos provistos por OpenAI.

Figura 8. Estimación de costos de uso del modelo de OpenAI

Usuarios estimados	Tiempo	Costo por token (input)	Costo por token (output)		
50	30	\$ 0.000012	\$ 0.000016		
Tokens de entrada esperados x día	Tokens de salida estimados x día	Costo estimado entrada	Costo estimado salida	Total	
500	1000	\$ 9.00	\$ 24.00	\$	33.00

7.2. Implementación de modelo LLM

El modelo a utilizar es un LLM que forma parte de los modelos desarrollados por OpenAI. Actualmente cuentan con 3 modelos (GPT-4, GPT-3.5 y GPT-3) capaces de entender y generar lenguaje natural. Cada uno de estos varía tanto en la cantidad de datos de entrenamiento, velocidad

de respuesta, cantidad máxima de *tokens* y capacidades de uso [7].

GPT-3 es un modelo que utiliza una variante de la arquitectura *transformer* conocida como *transformer decoder*, la cual se basa en generar una respuesta de lenguaje natural palabra por palabra de izquierda a derecha. A pesar de ser un modelo bastante eficiente, ha sido reemplazado por GPT-3.5; este, sumado a las tareas tradicionales de comprensión y generación de lenguaje natural, ha sido optimizado para el uso de completación de chat, a diferencia de GPT-3 el cual únicamente se utiliza como completación de texto. Finalmente está GPT-4, el cual es un modelo “capaz de resolver problemas difíciles con una mayor exactitud con respecto a los modelos anteriores” [7]. Si bien GPT-4 está basado en GPT-3, debido a que su entrenamiento consistió en un conjunto de datos más grande, esto conllevó a una mejora no solo en *performance*, sino en su capacidad de razonamiento gracias a un conocimiento general más amplio [38].

Si bien los LLM son modelos bastante eficientes, estos se describen como *pre-training models*, lo cual indica que son entrenados sobre una gran cantidad de datos; debido a esto, los modelos no son especializados en un tema particular. Es por ello que los modelos de LLM son capaces de ser *fine-tuned* o reentrenados sobre un conjunto de datos específico para desarrollar una tarea particular [62].

En los primeros prototipos del asistente se utilizaron los modelos *text-davinci-002* y *text-davinci-003*. Para implementar estos modelos se desarrolló una página *web* mediante *Streamlit*, la cual sirvió de interfaz para realizar pruebas con usuarios. El primer modelo (*text-davinci-002*) proporciona una respuesta rápida y corta, la cual toma alrededor de 5-10 segundos en completarse y la respuesta no es de más de 5 líneas. Por su parte, el otro modelo toma un poco más de tiempo en completar la respuesta, alrededor de 30-40 segundos, pero provee una respuesta más detallada, con *bullet points* sobre aspectos clave para la detección de un tipo de ataque. Los dos modelos se utilizaron en pruebas con usuarios para identificar cuál es el que más les gusta y/o comprenden.

Sin embargo, debido a que estos modelos son de la línea de completación de texto (es decir, únicamente toman una cadena de texto como entrada y con base en eso generan una respuesta), esto no siempre era lo ideal, debido a que en ocasiones los usuarios buscaban más información con respecto a un tema que anteriormente ya habían consultado. Es por ello que se optó por utilizar los modelos de completación de chat, los cuales, a diferencia de los de completación de texto, proveen una respuesta con base en un diálogo. Este tipo de modelo permite que el usuario pueda hacer referencia a mensajes anteriores dando una sensación de una conversación más humana.

7.3. Generación de datos de entrenamiento

Uno de los principales retos en la recopilación de información para hacer *fine tuning* del modelo es el hecho de que la mayoría de información se encuentra en inglés; sumado a esto, encontrar conjuntos de datos predefinidos con un dominio específico, en este caso de ciberseguridad, es muy complejo o al menos no existe en el ámbito *open source*. Por lo tanto, se optó por la generación manual de un conjunto de datos especializado en ciberseguridad.

Para la generación de este conjunto de datos se desarrolló un *web scraper* mediante Python; esto básicamente lo que hizo fue consumir datos de una página *web*, especialmente Wikipedia. Con esto se recopiló toda la información dentro de la página *web* y se concatenó en un solo *string* o párrafo que luego facilitó el formateo y unificación de los datos.

Figura 10. Implementación de dos modelos de GPT-3. Respuesta rápida *text-davinci-002* y respuesta lenta *text-davinci-003*

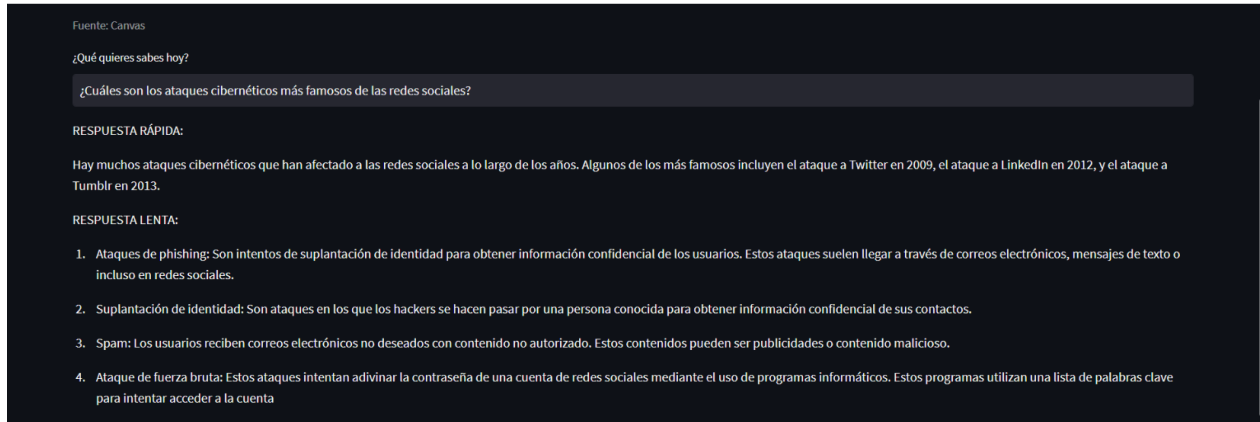


Figura 11. Comparación de respuesta de dos modelos de GPT-3 al consultar sobre un ataque profesional

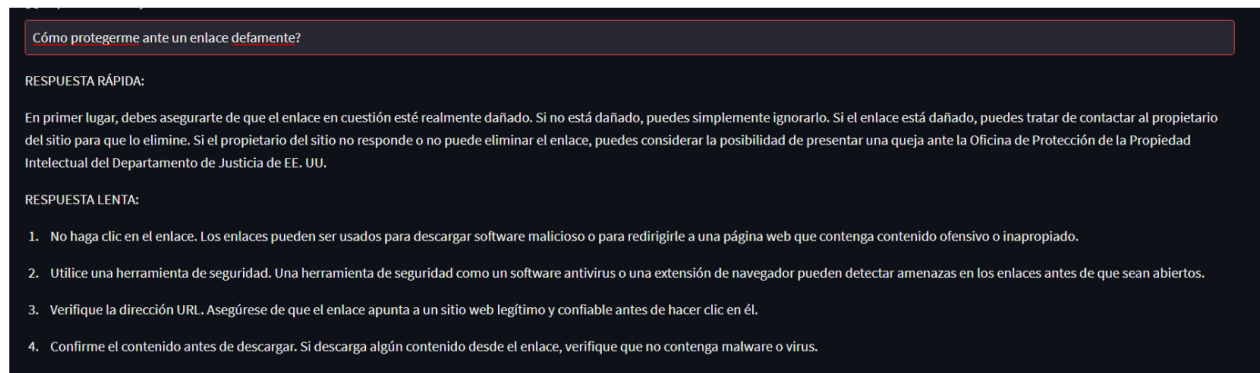
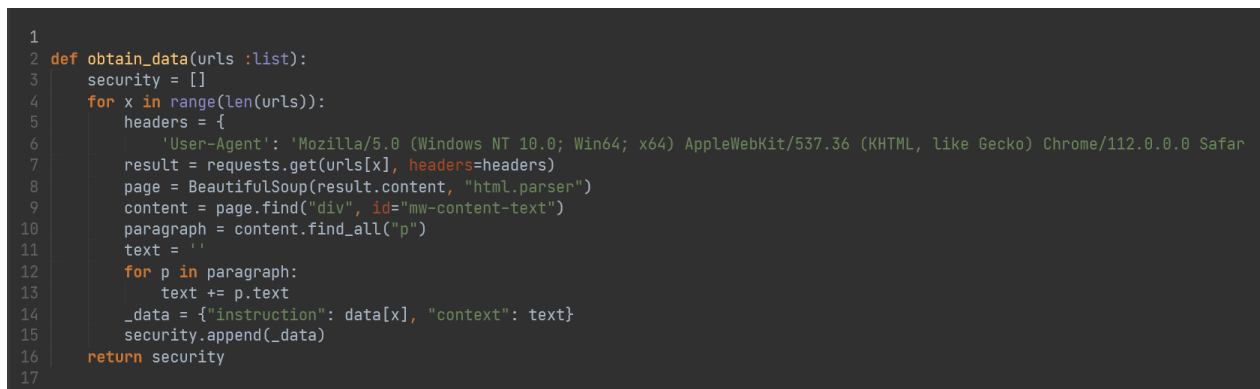


Figura 9. Desarrollo de un web scraper mediante Python



Algo que se debe tomar en cuenta es que, debido a que la posibilidad de hacer *fine tuning* del modelo de GPT-3.5 surgió luego de que ya se hubiera realizado una afinación en uno de los modelos de GPT-3, se realizaron dos procesos distintos de recopilación de información, los cuales se describen a continuación:

7.3.1. Conjunto de datos de entrenamiento para GPT-3

En el caso de los modelos de GPT-3, debido a que estos son utilizados para completación de texto, los datos de entrenamiento deben contener un *prompt* y un contexto, tal y como se muestra en la figura [12](#).

Figura 12. Ejemplo del formato de datos de entrenamiento para modelos de GPT-3

```
[
  {"prompt": "<prompt text>", "completion": "<ideal generated text>"},
  {"prompt": "<prompt text>", "completion": "<ideal generated text>"},
  {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
]
```

Donde *prompt* hace referencia a la consulta que realiza el usuario, y *completion* a lo que se espera que el modelo dé como respuesta.

Figura 13. Ejemplo de conjunto de datos de entrenamiento

```
[{'instruction': 'Qué es seguridad informática',
 'context': 'La seguridad informática , también conocida como ciberseguridad,[1]\u200b es el área relacionada con la info
{'instruction': 'Seguridad de la información',
 'context': 'La seguridad de la información es el conjunto de medidas preventivas y reactivas de las organizaciones y sist
{'instruction': 'Virus informático',
 'context': 'Un virus informático es un software que tiene por objetivo alterar el funcionamiento normal de cualquier tipo
{'instruction': 'Gusano informático',
 'context': 'Un gusano informático es un malware que se replica para propagarse a otras computadoras. Este software malici
{'instruction': 'Troyano',
 'context': 'En informática, se denomina caballo de Troya, o troyano, a un malware que se presenta al usuario como un prog
{'instruction': 'Bomba informática',
 'context': 'Una bomba lógica es una parte de un código insertado intencionalmente en un programa informático que permanec
{'instruction': 'Spyware',
 'context': 'EL programa espía[1]\u200b (en inglés spyware) es un malware que recopila información de una computadora y de
{'instruction': 'Malware',
 'context': 'Un programa malicioso (del inglés malware), también conocido como programa maligno, programa malintencionado
```

7.3.2. Conjunto de entrenamiento para GPT-3.5

Por su parte, en el caso de los modelos de GPT-3.5, al ser estos utilizados no únicamente para *text completion* sino también para chat, el formato de datos de entrenamiento de estos modelos es más parecido a una conversación, tal como se observa en la figura [14](#).

Figura 14. Ejemplo del formato de datos de entrenamiento para modelos de GPT-3.5

```
{
  "messages": [
    {"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."},
    {"role": "user", "content": "What's the capital of France?"},
    {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}
  ]
}
```

Tal y como se puede observar en la figura [14](#), se necesitan dos roles dentro de los datos: uno de ellos es el usuario (con el *prompt* que ingresaría como consulta) y el otro es el sistema, con la

respuesta correspondiente. A su vez, al sistema en general se le define una descripción, la cual le da un contexto base para proporcionar las respuestas.

Si bien en este caso también era posible utilizar el *web scraper* para obtener los datos generales para hacer el reentrenamiento, el proceso de clasificación y elaboración de este conjunto de datos requirió de más tiempo y esfuerzo. Esto debido a que era necesario hacer una lectura detallada de la información recopilada para luego formular el mensaje que el usuario pudiera ingresar.

A su vez, debido a que en este caso también se buscaba que el modelo fuera capaz de proporcionar *tips* cortos sobre ciberseguridad, así como datos interesantes de los mismos, dentro de los datos de entrenamiento se incluyeron algunos ejemplos de estos casos para que el modelo fuera capaz de retornar este tipo de información.

Figura 15. *Snippet de conjunto de datos utilizado para hacer fine tuning del modelo*

```
{
  "messages": [
    {
      "role": "system",
      "content": "Ela is a virtual assistant expert on cybersecurity"
    },
    {
      "role": "user",
      "content": "Dame un dato interesante sobre ciberseguridad"
    },
    {
      "role": "assistant",
      "content": "Sabías que los tipos de virus informáticos son: Recycler, Troyano, Bombas lógicas, Gusano, Hoax, y Joke."
    }
  ]
},
{
  "messages": [
    {
      "role": "system",
      "content": "Ela is a virtual assistant expert on cybersecurity"
    },
    {
      "role": "user",
      "content": "Dame un dato interesante sobre ciberseguridad"
    },
    {
      "role": "assistant",
      "content": "Sabías que un virus recycler consiste en crear un acceso directo de un programa y eliminar su aplicación original."
    }
  ]
},
}
```

7.4. *Fine-tuning*

El proceso de *fine tuning* consta en hacer pequeños cambios al modelo en busca de que funcione lo mejor posible [63]. Básicamente un modelo es entrenado sobre un mayor número de datos, buscando la especialización en un tema particular. Esto permite alcanzar un mayor resultado comparado con hacer *prompting*, ya que reduce la tendencia de alucinación del modelo y a su vez se disminuye la cantidad de *tokens* en la respuesta y en el *prompt*, minimizando así la latencia en las peticiones [64].

El proceso para realizar *fine tuning* de cualquiera de los modelos de OpenAI consta de los siguientes pasos:

- Preparar el conjunto de datos: esto consiste en generar archivos con el formato adecuado dependiendo del modelo de GPT a utilizar. Se debe tomar en consideración que cada uno de los datos de entrenamiento está limitado a 4096 *tokens*. Si la respuesta o el contenido excede esta cantidad, será limitado.
- Crear un *fine-tuning job*: este se crea mediante el SDK de OpenAI, y se le pasan los parámetros como los datos de entrenamiento y el nombre del modelo a entrenar. El entrenamiento puede durar desde minutos a horas, todo depende de la cantidad de datos de entrenamiento así como de la posición en la lista de trabajos.
- Utilizar el modelo recientemente *fine tuned*: una vez que el modelo haya terminado el proceso de entrenamiento, es posible utilizarlo mediante el nombre de identificación retornado en los detalles del entrenamiento.

El objetivo del *fine tuning* es especializar al modelo en el ámbito de la ciberseguridad, asegurando así que sus respuestas sean confiables y precisas. La figura [15](#) presenta un fragmento de los datos de entrenamiento utilizados.

Cabe resaltar que existen diversas estrategias mediante las cuales se puede lograr que un modelo sea capaz de aprender una tarea en particular; entre estas se encuentran:

- *Few-shot (FS)*: este término hace referencia al hecho de que el modelo recibe una cantidad reducida de ejemplos en los datos de entrenamiento (pero debe ser más de uno); sin embargo, en este caso, el modelo no actualiza los pesos con los cuales en su momento fue entrenado.
- *One-shot (1S)*: esta estrategia es muy similar a *few-shot* con la diferencia de que únicamente se da la descripción de la tarea que debe realizar, así como un ejemplo de la misma. El uso de esta u otra estrategia depende principalmente de cómo una u otra tarea sea solicitada a los humanos.
- *Zero-shot (0S)*: en este caso el modelo únicamente recibe una instrucción describiendo la tarea que debe realizar; no hay ningún ejemplo de lo que se espera recibir. Este tipo de estrategia tiene ciertas ventajas, como el hecho de que permite una mayor generalización de las tareas; sin embargo, es el método más complejo debido a que existen tareas que pueden parecer muy ambiguas a menos que se utilice un ejemplo [38](#).

Para el proceso de *fine tuning* se utilizó la estrategia de *one-shot*, ya que, tal como se puede observar en el ejemplo de datos de entrenamiento, se tiene únicamente un “ejemplo” por cada pregunta; es decir, para cada respuesta solo existe una única pregunta y viceversa. Finalmente, el proceso de *fine tuning* se realizó con un total de 32,407 *tokens* durante 3 épocas; una vez se realizaron pruebas a los datos para validar que el formato esté correcto y no exista ningún error o ejemplo que sobrepase el límite de *tokens*, se creó la tarea de entrenamiento mediante la API de OpenAI. Este proceso duró aproximadamente 3 horas y luego de eso se obtuvo un ID del nuevo modelo con mayor conocimiento en ciberseguridad.

Figura 16. Resultado del proceso de fine tuning al modelo de completación de chat de OpenAI

```
<FineTuningJob fine_tuning.job id=ftjob-0S5NtWUBgEBs1B8GxpYqBgUm at 0x2096a8753b0> JSON: {
  "object": "fine_tuning.job",
  "id": "ftjob-0S5NtWUBgEBs1B8GxpYqBgUm",
  "model": "gpt-3.5-turbo-0613",
  "created_at": 1697899897,
  "finished_at": 1697901464,
  "fine_tuned_model": "ft:gpt-3.5-turbo-0613:ela:cybersecurity-ela:8C81S6j3",
  "organization_id": "org-Zg0RSdq1kmqgWqtS6QZM0nB0",
  "result_files": [
    "file-In3EvM5lFNfoUxtBQrSvlsNU"
  ],
  "status": "succeeded",
  "validation_file": null,
  "training_file": "file-swfICSv2Jxqi28F1bnhq9qj7",
  "hyperparameters": {
    "n_epochs": 3
  },
}
```

7.5. Desarrollo de modelo de clasificación de dominios

Una de las funcionalidades principales del asistente virtual es la capacidad de poder detectar dominios maliciosos. Sin embargo, el modelo de generación de lenguaje natural no es capaz de realizar este proceso de clasificación. Es por ello que se desarrolló un modelo independiente encargado de la clasificación de dominios.

Los atacantes generalmente en la creación de dominios maliciosos suelen utilizar varias técnicas de ofuscación y *typosquatting* para hacer que estos parezcan seguros. Ofuscación básicamente es hacer que algo parezca menos claro o difícil de comprender de manera intencional [65] y *typosquatting* se refiere a registrar un dominio muy similar al de un sitio *web* conocido pero con algún tipo de error de escritura [66]. Con base en lo anterior, el modelo de clasificación de dominios maliciosos se desarrolló mediante la recolección de características léxicas y estadísticas del dominio.

Las características léxicas corresponden a todos aquellos *features* extraíbles directamente del nombre del dominio, tales como la longitud, la cantidad de caracteres numéricos o la cantidad de símbolos. De igual forma, en el caso de las características estadísticas, estas se conforman por la información de *WHOIS* provista por la respuesta del servicio de DNS (*Domain Name Service*), el cual es el encargado de asignar un nombre de dominio y relacionarlo a una IP.

Para el proceso de entrenamiento del modelo se obtuvo una serie de dominios, derivados de diversas fuentes detalladas en la sección de *Responsible AI*, que correspondían a clases distintas (benignos, *malware*, *phishing* y *ransomware*). El conjunto de datos original se conformaba de 458,000 dominios de *phishing*, 1 millón de dominios correspondientes a la clase benigna, 1,904 clasificados como *ransomware* y 435,405 dominios de *malware*.

Para buscar balancear los datos, inicialmente se realizó un muestreo, dejando como límite superior medio millón de datos. Luego de esto se buscó obtener la información estadística de los dominios

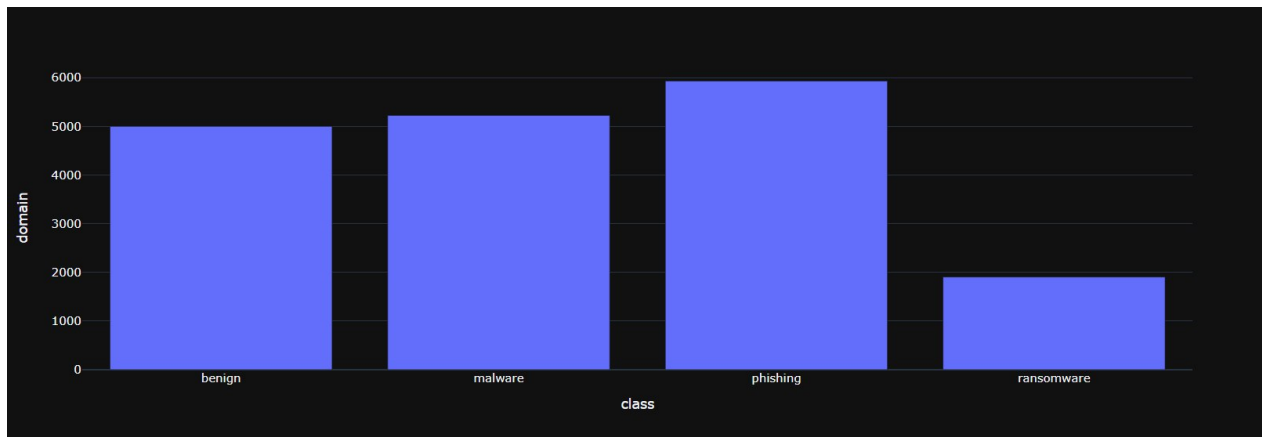
mediante consultas al servidor *WHOIS* con uso de una librería en Python. En el proceso de consulta de los dominios benignos a *WHOIS* se observó que luego de haber pasado aproximadamente 3 horas, únicamente se había obtenido información de alrededor del 1% de los datos. Esto implicaba que para poder obtener la información de 500,000 datos se iba a tardar alrededor de 8 días; esto, por los cuatro conjuntos de datos, significaba una duración aproximada de 25 días.

Por lo anterior, se decidió realizar un nuevo muestreo aleatorio, seleccionando un porcentaje de los datos que permita obtener alrededor de 5000 dominios de cada clase, buscando así reducir el tiempo de recopilación de *features*. Cabe destacar que de la categoría de *ransomware* únicamente se contaban con 1904 dominios, generando así un conjunto de datos no balanceado.

Al calcular el grado de desbalance en las clases, se obtuvo una relación 1:3; es decir, por cada dominio de *ransomware* existen 3 dominios de *phishing*, *malware* o benignos. En este caso se considera un desbalance “ligero”, ya que las muestras del conjunto de datos están desiguales por una pequeña cantidad de datos. Este tipo de desbalanceo generalmente no se suele considerar un problema para los modelos de clasificación debido a que no representa una desventaja significativa para la clase desbalanceada en cuanto al aprendizaje de características mediante el modelo [67]. Es por ello que el conjunto de datos final cuenta con:

- 5,000 dominios benignos
- 5,750 dominios de *phishing*
- 5,300 dominios de *malware*
- 1,904 dominios de *ransomware*

Figura 17. Distribución de datos de entrenamiento para modelo de clasificación de dominios



Posteriormente, se realizó una selección de las características pertinentes para el entrenamiento del modelo. Se utilizaron tanto características léxicas como estadísticas. Una de las características estadísticas más predominantes fue la duración o “vida” de un dominio, ya que mediante este parámetro es posible clasificar el tipo de ataque al cual corresponde el dominio [68]. A su vez, los dominios

Cuadro 1. Características utilizadas para entrenamiento del modelo de clasificación de dominios

Característica	Descripción
<i>lifetime</i>	Duración del dominio en años (fecha de expiración - fecha de creación)
<i>active-lifetime</i>	Espacio de tiempo en el cual el dominio se ha encontrado activo en días (fecha de edición - fecha de creación)
<i>amount-ips</i>	Cantidad de direcciones IP asociadas
<i>length</i>	Cantidad de caracteres que forman el dominio, incluyendo puntos
<i>subdomains</i>	Cantidad de subdominios asociados al Top Level Domain
<i>count-numbers</i>	Cantidad de números que contiene el dominio
<i>count-letters</i>	Cantidad de letras que contiene el dominio
<i>consecutive-numbers</i>	Cantidad de números consecutivos (número anterior es igual al siguiente)
<i>consecutive-letters</i>	Cantidad de letras consecutivas
<i>consecutive-symbols</i>	Cantidad de símbolos consecutivos

que han sido recientemente registrados suelen ser muy utilizados para ataques cibernéticos y suelen durar muy poco tiempo activos, lo que hace que su detección sea considerablemente compleja [69].

En la figura 18 se puede observar cómo el tiempo activo (medido en días) de un dominio representa una característica diferencial en el tipo de dominio. Se puede notar que los dominios benignos suelen durar en promedio alrededor de 5,000 días, mientras que el promedio de días activos de los dominios de *malware* y *phishing* está en alrededor de 1,200 y finalmente en el caso de los dominios de *ransomware* donde el promedio de días se encuentra alrededor de 30.

El siguiente paso para la construcción del modelo fue la selección de la arquitectura del mismo. En este caso, debido a que se busca que el modelo sea usado directamente dentro de dispositivos Android, se optó por utilizar TensorFlow, la cual es una herramienta para la construcción de modelos de redes neuronales. Esto con el propósito de optimizar recursos, tanto en almacenamiento como en CPU, gracias a que TensorFlow cuenta con un módulo desarrollado especialmente para la integración con aplicaciones Android, en el cual se asegura que los modelos sean lo suficientemente livianos para que puedan ser ejecutados sin problema.

Para el modelo desarrollado se utilizaron 8 capas, dentro de las cuales 7 de ellas eran capas “ocultas” y la última la capa de salida. Dentro de las capas ocultas, se contó con 4 capas densas, las cuales se constituyeron de 256, 64, 32 y 16 neuronas respectivamente. A su vez, entre cada una de las capas densas se agregaron capas regularizadoras de *dropout* para evitar la especialización en las características de los datos de entrenamiento. Por su parte, la capa de salida se conformó de 4 neuronas, ya que eran 4 tipos de clases los cuales debía de clasificar, y para realizar la predicción se utilizó la función de activación sigmoide. Esta función de activación retorna un vector de probabilidades, asignando a cada una de las clases una probabilidad de que el dominio ingresado para clasificación pertenezca a esa clase. Finalmente, para la clasificación del dominio, se obtiene la clase con mayor probabilidad.

Figura 18. Promedio de tiempo activo de un dominio



Figura 19. Porcentaje de símbolos promedio en cada uno de los tipos de dominio

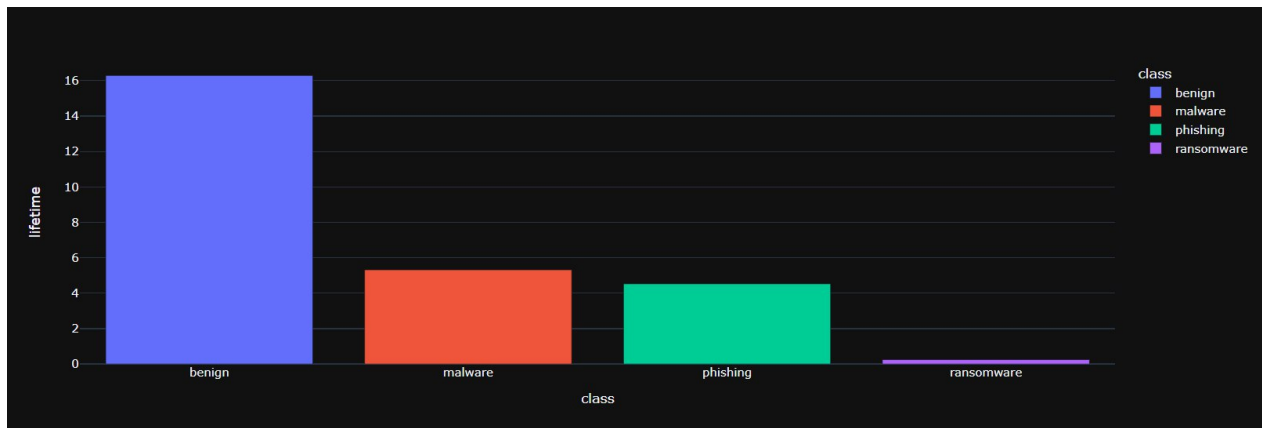


Figura 20. Cantidad promedio de letras consecutivas para cada tipo de dominio



Figura 21. *Arquitectura del modelo de clasificación de dominios*

```
model = models.Sequential([
    layers.InputLayer(input_shape=(x_train.shape[1],)),
    layers.Dense(256, activation="relu"),
    layers.Dropout(0.1),
    layers.Dense(128, activation="relu"),
    layers.Dropout(0.1),
    layers.Dense(64, activation="relu"),
    layers.Dropout(0.1),
    layers.Dense(16, activation="relu"),
    layers.Dropout(0.2),
    layers.Dense(4, name='out_layer', activation="sigmoid")
])

model.compile(optimizer=optimizers.Adam(learning_rate=0.00022), loss='categorical_crossentropy', metrics=['accuracy'])
Executed at 2023.11.16 17:32:41 in 75ms
```

7.6. Pruebas de conocimiento e interacción con el asistente virtual

Para realizar una serie de pruebas que permitan evaluar el conocimiento que los usuarios pueden llegar a adquirir al utilizar un asistente virtual centrado en ciberseguridad, se integró el modelo LLM *fine tuned* y el modelo de clasificación de dominios en una aplicación denominada ELA [70].

El proceso de pruebas se llevó a cabo en dos fases: la primera se centró en el uso de la aplicación para la interacción con el asistente virtual, y la segunda en la simulación de pruebas de *phishing* controladas con los usuarios de prueba.

Antes de participar, los usuarios recibieron información detallada sobre el propósito de las pruebas, los datos que se recopilarán y cómo se utilizarán. Se les proporcionó un documento de consentimiento informado que firmaron antes de participar en el estudio (ver Anexos).

Se realizaron dos pruebas: la primera se denominó prueba preliminar, ya que en este caso se buscaba obtener información sobre el funcionamiento de la aplicación en conjunto con los modelos, así como retroalimentación de los usuarios sobre posibles errores. Y la segunda prueba, una prueba piloto, la cual buscaba evaluar la funcionalidad de ELA para los usuarios.

Para ambas pruebas se implementó un cuestionario; en el caso de la preliminar únicamente se les compartió antes del uso de ELA. Por su parte, en el caso de las pruebas piloto, el cuestionario se les compartió a los participantes antes y después de la interacción con el asistente. Esto para evaluar el nivel de conocimiento de los participantes sobre ciberseguridad y medir si con el uso del asistente los participantes son capaces de generar nuevo conocimiento acerca de ataques cibernéticos y cómo protegerse.

Para la prueba piloto se compartieron dos versiones de la aplicación: una versión *explain*, la cual al momento de bloquear algún dominio malicioso proporciona detalles sobre el tipo de ataque y cómo protegerse ante este ataque; y una versión *no explain*, la cual únicamente realiza el bloqueo del dominio y consulta al usuario si este desea permitir la navegación a la página *web*.

Durante el periodo de prueba (tanto preliminar como piloto) se motivó a los participantes a utilizar la aplicación y el asistente virtual haciendo consultas de manera regular (5 preguntas a la semana). Luego, cuando los usuarios de prueba ya se encontraban familiarizados con el asistente, es

decir ya lo habían utilizado por alrededor de 1 semana y media, se enviaron pruebas simuladas de *phishing* para evaluar el impacto en la ciberseguridad de los usuarios al utilizar el asistente.

7.7. Simulación de ataque de *phishing*

Para poder determinar si una persona es menos propensa a ser víctima de un ataque cibernético mediante la explicación de los tipos de *malware* que utiliza un dominio, se desarrollaron una serie de pruebas de ataques cibernéticos.

El proceso de creación de estas pruebas fue el siguiente:

- Selección de tipos de ataque cibernético: se seleccionó uno de los ataques más populares en la actualidad, *phishing*, en particular *spear phishing*. En este tipo de ataque, el atacante primero obtiene información sobre la posible víctima (nombre, gustos, mascotas, familia) para así poder dar la sensación de familiaridad.
- Selección de vector de ataque: se decidió utilizar el correo electrónico debido a la alta tasa de correos de *phishing* recibidos globalmente y, a su vez, en busca de un vector de ataque con un acceso más inmediato a los usuarios, se utilizó la mensajería de texto.
- Selección de servicio/empresa a suplantar: se decidió utilizar Facebook debido a la gran popularidad de la red social y a la alta familiaridad de las personas con una red tan conocida. De igual forma, se utilizó Spotify debido a que todos los participantes contaban con una cuenta de Spotify y al menos un 75 % la utilizaba de manera regular.

Posterior a esto se preparó todo el material audiovisual necesario para las pruebas. En el caso de Facebook se desarrolló una simulación de un cambio de contraseña; por lo tanto, fue necesario una copia de un correo electrónico enviado por Facebook al momento de hacer un cambio de contraseña, al igual que una copia de su página de inicio de sesión.

A su vez se desarrolló un inicio de sesión de Spotify y en este caso el correo indicaba sobre un error en el método de pago del servicio.

Figura 22. Imitación de página de inicio de sesión de Spotify

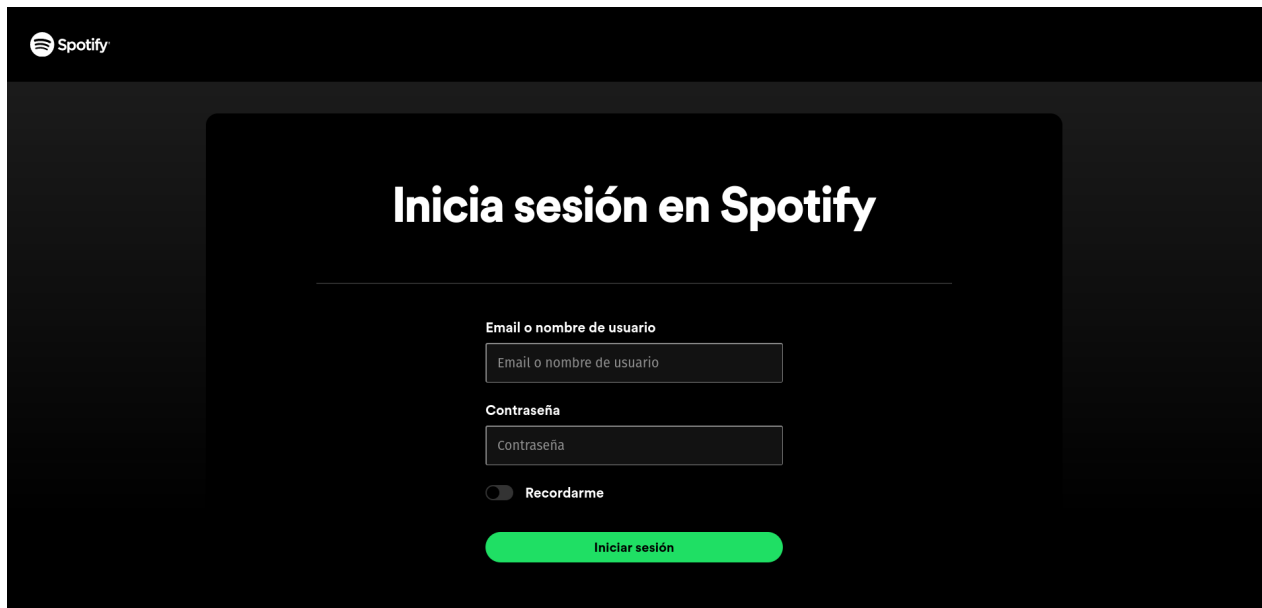


Figura 23. Copia de página de login de Facebook para prueba de simulación de phishing

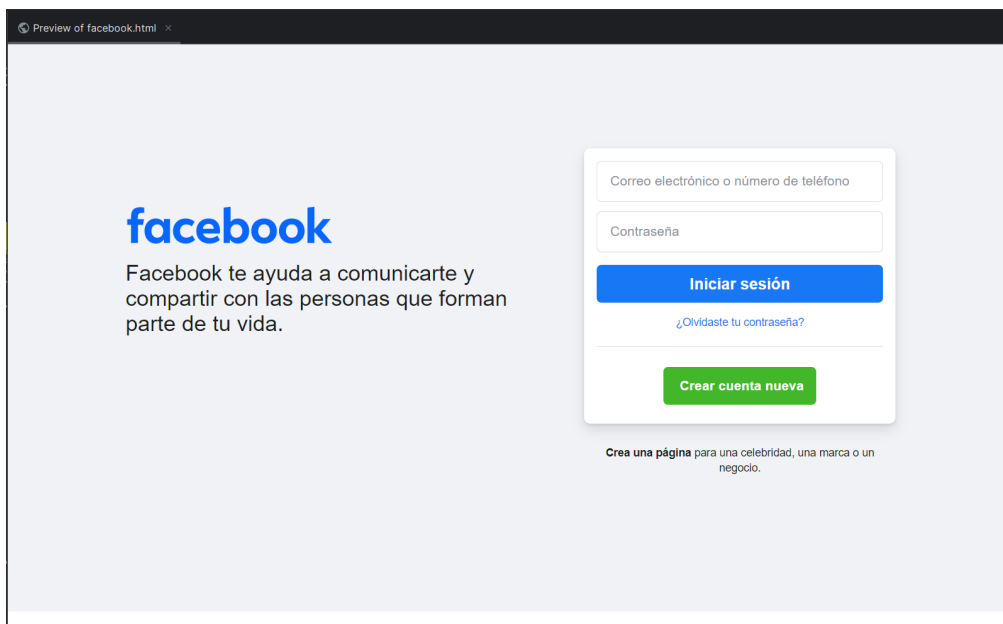
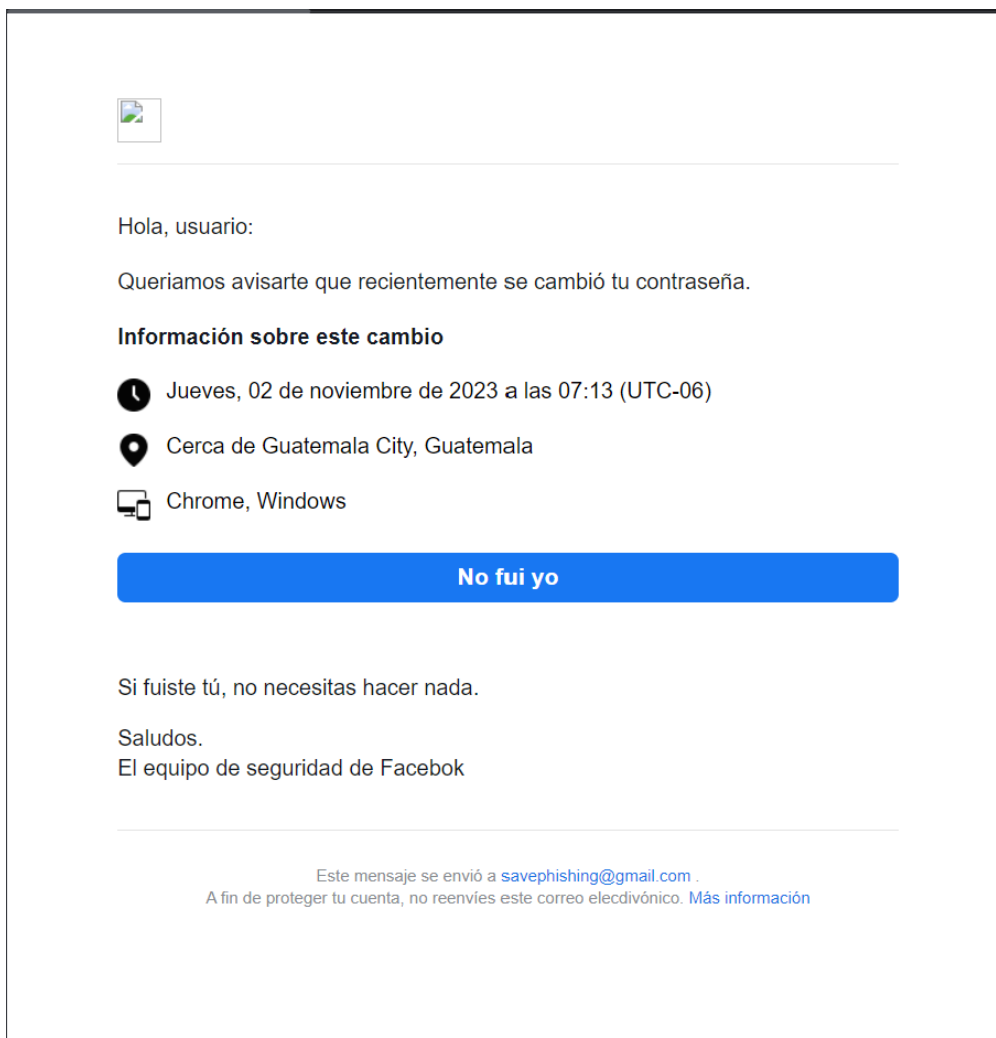


Figura 24. Copia de correo electrónico enviado por Facebook al momento de cambiar contraseña



Los resultados se componen de cuatro etapas principales: un nuevo modelo de LLM *fine tuned* en ciberseguridad, métricas del modelo de clasificación de dominios, interacción de usuarios con modelo LLM y modelo de clasificación, y la simulación de prueba de ataques de *phishing*.

8.1. Pruebas de prototipo

Las pruebas con usuarios del prototipo desarrollado, utilizando los modelos de LLM text-davinci-002 y text-davinci-003, involucraron la participación de 7 usuarios. El propósito de estas pruebas (ver Anexos) fue obtener una comprensión de la percepción de los usuarios con respecto al tiempo de respuesta del sistema (rápido o lento) y determinar en qué escenarios específicos preferían cada tipo de respuesta.

Como resultado de estas pruebas, se obtuvo lo siguiente:

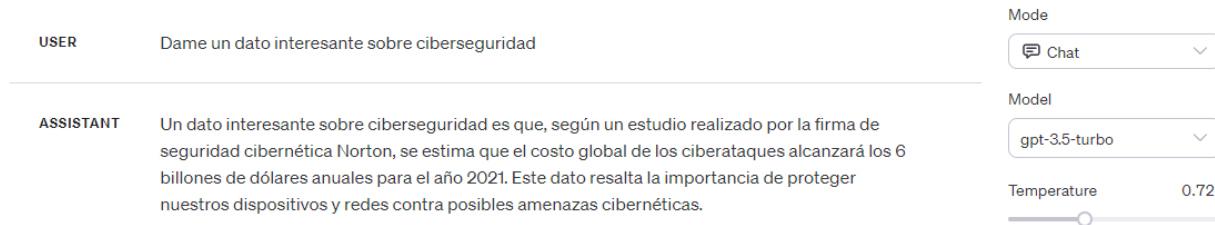
- Cuando los usuarios buscaban aprender sobre cómo protegerse ante un ataque particular, 5 de cada 7 usuarios preferían la respuesta larga debido a que proveía de manera enumerada una serie de *tips* y consejos, a diferencia de la respuesta corta la cual únicamente respondía un párrafo sobre el principal consejo de protección.
- Cuando los usuarios buscaban aprender la definición de algún tipo de virus o ataque, 6 de cada 7 preferían la respuesta corta que les permitiera aprender de manera concreta sobre el tipo de ataque que estaban consultando.

Con base en estas solicitudes se desarrollaron los datos de entrenamiento para hacer *fine tuning* del modelo y así poder proveer pasos enumerados para la protección contra un ataque y, a su vez,

generar una definición concisa, con pocas palabras y lenguaje no muy técnico, para facilitar el aprendizaje.

A su vez, en busca de proveer una idea de conocimiento diario, también se agregaron a los datos de entrenamiento de *fine tuning tips* diarios, que de manera corta puedan proveer un dato interesante sobre ciberseguridad, especialmente definiciones de ataques o *tips* de cómo protegerse y navegar de manera segura.

Figura 25. Solicitud de un dato interesante de ciberseguridad al modelo GPT-3.5 de OpenAI

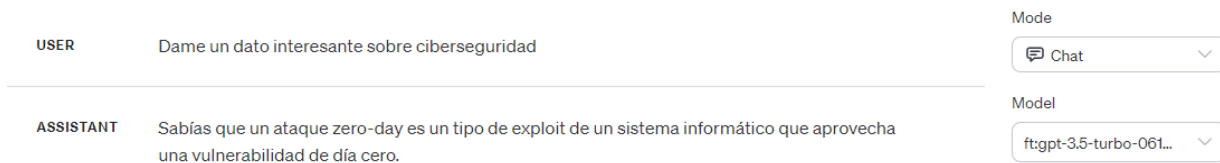


8.2. Modelo especializado

Una vez se obtuvo el identificador del nuevo modelo *fine-tuned*, se realizaron comparaciones con la versión normal de GPT-3.5 para observar las diferencias en cuanto a las respuestas provistas de ciberseguridad.

Por ejemplo, en las figuras 25 y 26 se tiene una comparación de la respuesta de dos modelos ante el mismo *prompt* o entrada ingresada por el usuario. Tal como se observa, en el caso del modelo GPT-3.5, este simplemente provee algún dato relevante sobre ciberseguridad, mientras que el modelo *fine tuned* retorna información corta sobre un tipo de ataque cibernético.

Figura 26. Solicitud de un dato interesante sobre ciberseguridad al modelo *fine tune*



En las figuras 27 y 28 se pueden contrastar las respuestas otorgadas mediante cada uno de los modelos proporcionados; en este caso se observa que la respuesta propuesta por GPT-3.5, además de ser más extensa, en los consejos para protegerse del tipo de ataque también agrega una pequeña explicación ampliando dicho consejo. Mientras que, por su parte, en el caso del modelo *fine tuned* la respuesta provista es bastante directa y no cuenta con una explicación extra con respecto a cada *tip* porque busca que el usuario sea capaz de leer los consejos e implementarlos de manera rápida.

Figura 27. *Solicitud de información sobre un ataque mediante el modelo de GPT-3.5*

USER	Qué es phishing y cómo puedo evitar ser víctima de phishing?
ASSISTANT	<p>El phishing es un tipo de estafa en línea en la que los delincuentes se hacen pasar por entidades legítimas para obtener información personal y confidencial de los usuarios, como contraseñas, números de tarjetas de crédito, información bancaria, entre otros.</p> <p>Para evitar ser víctima de phishing, puedes seguir estas recomendaciones:</p> <ol style="list-style-type: none">1. Sé cauteloso con los correos electrónicos y mensajes sospechosos: No hagas clic en enlaces ni descargues archivos adjuntos de mensajes de remitentes desconocidos o sospechosos.2. Verifica la autenticidad de los sitios web: Antes de ingresar información personal o financiera en un sitio web, asegúrate de que la dirección comience con "https://" y de que haya un candado en la barra de direcciones.3. No compartas información personal o financiera a través de correos electrónicos: Las entidades legítimas nunca solicitarán información confidencial por correo electrónico. Evita proporcionar datos sensibles a través de este medio.4. Mantén actualizados tus dispositivos y programas: Los sistemas operativos, navegadores y programas de seguridad deben estar actualizados para protegerse contra las últimas amenazas de phishing.5. Utiliza software de seguridad confiable: Instala y actualiza regularmente un programa antivirus o antimalware en tu dispositivo para detectar y bloquear posibles ataques de phishing.

Figura 28. Solicitud de información sobre un ataque mediante el modelo fine tuned

The image shows a chat interface with a user query and an assistant's response, alongside a control panel for the model.

USER ¿Qué es phishing y cómo puedo protegerme ante ataques de phishing?

ASSISTANT Phishing es un tipo de fraude por Internet que busca obtener información personal, claves de acceso, información de tarjetas de crédito, u otros datos sensibles de forma fraudulenta.
La mejor forma de protegerse de ataques de phishing es:
- No dar información personal o financiera a nadie por correo electrónico.
- No hacer clic en enlaces en mensajes de correo electrónico que te soliciten información personal.
- No abrir archivos adjuntos en mensajes de correo electrónico que te soliciten información personal.
- No hacer clic en enlaces en mensajes de correo electrónico que te pidan actualizar información personal.
- Ignorar los correos electrónicos de factura que indican que se le cobrará por una suscripción o servicio si no cancela "la orden" escribiendo a la empresa.
- No hacer clic en el link de cancelación de suscripción de un mensaje de correo electrónico, a menos que sea una suscripción a la que te suscribiste al 100%.
- Nunca hagas clic en un enlace en un mensaje de correo electrónico para ir a sitios web como tu banco. En su lugar, escriba la URL directamente en el navegador web. Si el mensaje afirma ser del banco, llámalo, pero no uses el número de teléfono proporcionado en el mensaje.

Model ft:gpt-3.5-turbo-06...
Temperature 0.75
Maximum length 673
Stop sequences Enter sequence and press Tab
Top P 1
Frequency penalty 0
Presence penalty 0
API and Playground requests will not be used to train our

Figura 29. *Solicitud de información sobre un ataque mediante el modelo fine tuned con datos de entrenamiento formateados*

USER	¿Qué es phishing y cómo puedo protegerme ante ataques de phishing?
ASSISTANT	<p>Phishing es un tipo de delito informático y estafa en la que un atacante cibernético, suplanta una identidad o utiliza técnicas de ingeniería social para engañar a una víctima y así obtener dinero, información confidencial o acceso a información y/o sistemas.</p> <p>Para protegerte ante este tipo de ataques se recomienda:</p> <ul style="list-style-type: none">- No brindar información personal o confidencial a través de correos electrónicos o mensajes de texto.- No acceder a enlaces que soliciten la actualización de datos personales, especialmente a servicios bancarios o en correos electrónicos sospechosos.- Cerciórate de la dirección web de la página donde se solicita la información personal, si es una dirección sospechosa o no corresponde al servicio, no accedas.- No acceder a enlaces en correos electrónicos o mensajes de texto que parecen provenir de servicios que requieren que ingreses información personal, especialmente si el contenido del mensaje es alarmante o amenazador.- Utilizar servicios de correo que identifiquen spam y phishing automáticamente y notifiquen al usuario de la presencia de estos mensajes.- Utilizar servicios que verifiquen la reputación de un sitio web o enlace antes de acceder al mismo, especialmente si estos provienen de correos electrónicos o mensajes de texto.

8.3. Modelo de clasificación de dominios

El modelo de clasificación de dominios fue entrenado mediante la función de optimización Adam, con una tasa de aprendizaje de 0.00022. Se entrenó durante 150 épocas y se alcanzó una exactitud (*accuracy*) con los datos de validación de 0.7; esto indica que al menos en el 70% de los casos es capaz de detectar correctamente un dominio.

Figura 30. Métricas durante el proceso de entrenamiento del modelo de clasificación de dominios

```
history = model.fit(x_test, to_categorical(y_test), epochs=150, validation_split=0.16, verbose=2, batch_size=32)
Executed at 2023.11.03 03:37:38 in 2m 36s 826ms
-----
143/143 - 1s - loss: 0.7062 - accuracy: 0.6955 - val_loss: 0.7007 - val_accuracy: 0.6993 - 1s/epoch - 8ms/step
Epoch 146/150
143/143 - 1s - loss: 0.7045 - accuracy: 0.6933 - val_loss: 0.6947 - val_accuracy: 0.6993 - 1s/epoch - 8ms/step
Epoch 147/150
143/143 - 1s - loss: 0.7076 - accuracy: 0.6974 - val_loss: 0.6912 - val_accuracy: 0.7016 - 598ms/epoch - 4ms/step
Epoch 148/150
143/143 - 1s - loss: 0.7137 - accuracy: 0.7001 - val_loss: 0.6923 - val_accuracy: 0.7051 - 510ms/epoch - 4ms/step
Epoch 149/150
143/143 - 1s - loss: 0.7067 - accuracy: 0.6987 - val_loss: 0.6955 - val_accuracy: 0.7085 - 716ms/epoch - 5ms/step
Epoch 150/150
143/143 - 1s - loss: 0.7020 - accuracy: 0.7064 - val_loss: 0.6900 - val_accuracy: 0.7085 - 975ms/epoch - 7ms/step
```

En el proceso de entrenamiento se utilizaron las curvas de aprendizaje para variar la tasa de aprendizaje del modelo y permitir que este fuera capaz de aprender de manera progresiva las características de los datos. A su vez, mediante el gráfico de curvas de aprendizaje se determinó la cantidad máxima de épocas, para evitar que el modelo sufriera de sobreentrenamiento. En la figura 31 se puede observar el gráfico de curva de aprendizaje del modelo, la cual presenta un comportamiento exponencial a lo largo de las épocas, indicando que a medida que incrementa el número de épocas el error en las predicciones disminuye.

Figura 31. Curva de aprendizaje de entrenamiento de modelo de detección de dominios maliciosos

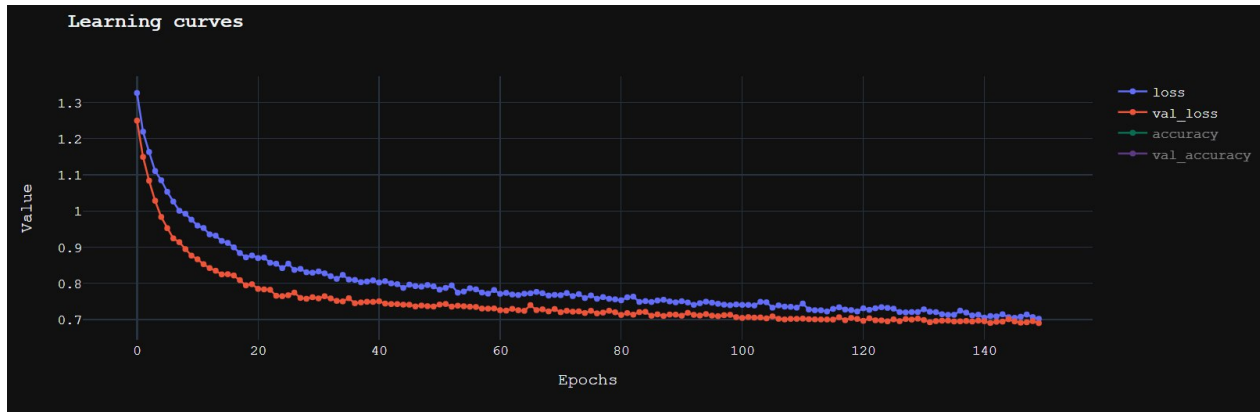
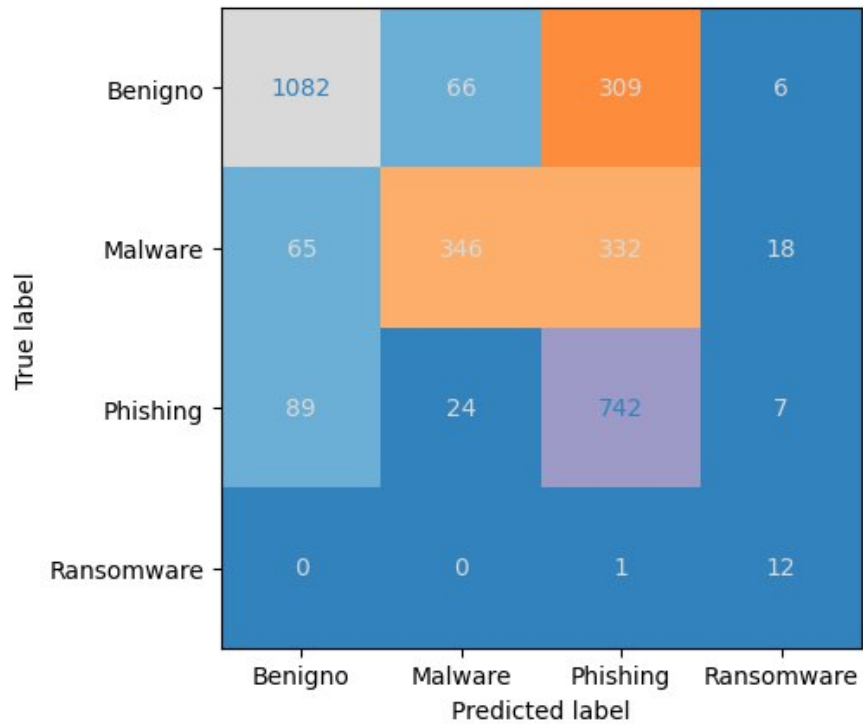


Figura 32. Reporte de clasificación del modelo de clasificación de dominios con el conjunto de prueba

	precision	recall	f1-score	support
Benigno	0.88	0.74	0.80	1463
Malware	0.79	0.45	0.58	761
Phishing	0.54	0.86	0.66	862
Ransomware	0.28	0.92	0.43	13
accuracy			0.70	3099
macro avg	0.62	0.74	0.62	3099
weighted avg	0.76	0.70	0.71	3099

Figura 33. Matriz de confusión del modelo de clasificación de dominios



Para evaluar el modelo en el conjunto de *test* se buscó simular los porcentajes de tráfico diario

en la vida real; es por ello que para el conjunto de prueba los datos se encuentran desbalanceados. Como se puede observar en la figura 32, la cual contiene el reporte de clasificación, al igual que con el conjunto de validación, con el conjunto de prueba se obtuvo un 70 % de exactitud.

Dentro de este reporte se observa cómo la precisión del modelo al clasificar dominios benignos es de 0.88, lo cual indica que del total de predicciones benignas realizadas por el modelo, un 88 % de estas fueron correctas. Esto es especialmente representativo debido a la índole del problema de clasificación, ya que si bien en general se busca que un modelo tenga alto *accuracy*, en este caso es mucho más importante reducir la cantidad de dominios maliciosos que el modelo predice como benignos; esto se puede medir a través de la precisión de dominios benignos y a su vez, en la matriz de confusión (figura 33). En general, se busca que la matriz de confusión para este modelo tenga la menor cantidad de falsos negativos posibles.

Por su parte, en las métricas también se observa que el modelo no logra distinguir correctamente entre los distintos tipos de dominios maliciosos. Ya que del total de dominios de *malware* dentro del conjunto de prueba, el modelo únicamente predijo correctamente un 45 % y el restante en su mayoría los predijo como *phishing*. Esto indica que el modelo confunde principalmente los dominios de *malware* con *phishing*. Se considera que ocurre debido a que, dentro de estas dos categorías, las características utilizadas para entrenamiento no eran significativamente diferentes, ya que por ejemplo, en la figura 18 se puede notar que la variación entre el tiempo activo promedio de un dominio de tipo *malware* y un dominio de *phishing* es bastante pequeña.

8.4. Pruebas de conocimiento e interacción con el asistente virtual

8.4.1. Pruebas preliminares

En la prueba preliminar se tuvo un total de 4 participantes, dentro de los cuales uno de ellos era adolescente, otros dos jóvenes y un adulto. Al inicio de la prueba se les compartió ELA y se les solicitó que lo instalaran e interactuaran con la aplicación.

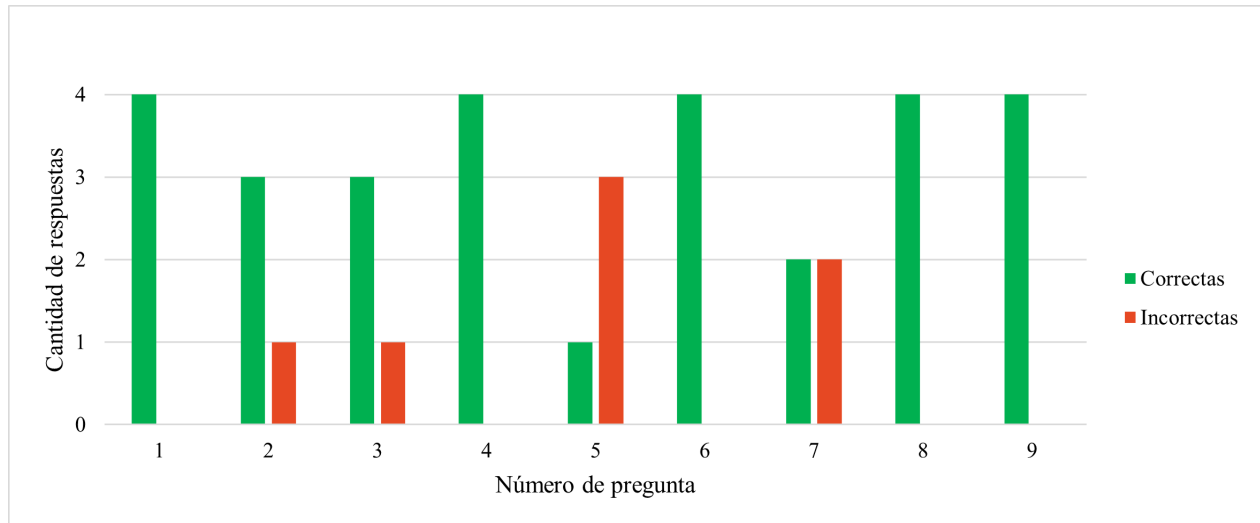
Luego de esto se les compartió un formulario mediante Google *Forms* de conocimientos básicos. Este cuestionario buscaba establecer una línea base acerca de qué tanto sabían los usuarios sobre algunos tipos de ataques famosos o técnicas para protegerse. Algo que comentaron los participantes es que, si bien estos no sabían la respuesta a alguna de las preguntas, ninguna de las preguntas tenía una opción de “no sé”.

La recopilación de las respuestas del formulario fue un proceso interesante, debido a que se obtuvieron resultados inesperados: todas las personas tenían alto conocimiento en ciberseguridad. Sin embargo, al consultar con uno de los participantes, este indicó que había hecho el cuestionario con ayuda de ELA (aplicación que implementa el asistente y el modelo de clasificación); es por ello que la mayoría de sus respuestas eran correctas. Esto evidentemente evitó establecer la línea base de conocimientos, ya que en general las personas mostraban un alto dominio de temas básicos de ciberseguridad (ver figura 34); sin embargo, permitió que las personas pudieran tener una mayor interacción con el asistente y en el proceso adquirir conocimiento.

Cuadro 2. Asociación entre número de pregunta en gráfica y pregunta en cuestionario para prueba preliminar

Pregunta	Descripción
1	¿Cuál es el tipo de ciberataque en el cual una persona intenta engañar a otras para que revelen información confidencial como contraseñas o números de tarjetas de crédito?
2	A Ana le llegó un correo aparentemente inofensivo que tenía como adjunto un PDF. Al descargar el PDF, se percató de que el ícono de sus archivos cambió a un candado. Luego recibió un correo indicando que toda su información había sido encriptada y necesitaba pagar Q100,000 para recuperarlos. Este es un ejemplo de:
3	¿Cuál de las siguientes medidas es una forma efectiva de protegerse contra <i>malware</i> ?
4	¿Cuál de las siguientes afirmaciones describe mejor la ingeniería social?
5	¿Cuál de las siguientes prácticas es una medida de seguridad efectiva para protegerse contra el robo de identidad?
6	¿Qué es el <i>vishing</i> ?
7	¿Qué es el doble factor de autenticación (2FA)?
8	¿Por qué es importante mantener las aplicaciones actualizadas?

Figura 34. Gráfico de respuestas a cuestionario de ciberseguridad antes de utilizar ELA en prueba preliminar



8.4.2. Pruebas piloto

Para las pruebas piloto se tuvo un total de 12 participantes, de los cuales el 75 % eran jóvenes adultos y el 25 % eran adultos con una edad mayor a 30. A diferencia de las pruebas preliminares, en este caso, antes de compartirlas la aplicación, primero se les compartió el cuestionario de conocimiento

Cuadro 3. Asociación entre número de pregunta en gráfica y pregunta en cuestionario para prueba piloto

Pregunta	Descripción
1	¿Cuáles de estas contraseñas consideras que son seguras?
2	A Ana le llegó un correo aparentemente inofensivo que tenía como adjunto un PDF. Al descargar el PDF, se percató de que el ícono de sus archivos cambió a un candado. Luego recibió un correo indicando que toda su información había sido encriptada y necesitaba pagar Q100,000 para recuperarlos. Este es un ejemplo de:
3	Recibiste un correo electrónico informándote que alguien accedió a tu cuenta de banco. Para poder bloquear este acceso, debes de proporcionar tus datos personales de manera urgente. ¿Qué decides hacer?
4	¿Cuál de los siguientes es un ejemplo de autenticación de dos pasos?
5	¿Qué es el <i>vishing</i> ?
6	¿Cuál de las siguientes afirmaciones describe mejor la ingeniería social?
7	Si debes hacer una cuenta en una página que no se ve segura. ¿Qué decides hacer?
8	¿Cuál de las siguientes prácticas es esencial para realizar copias de seguridad?

en ciberseguridad.

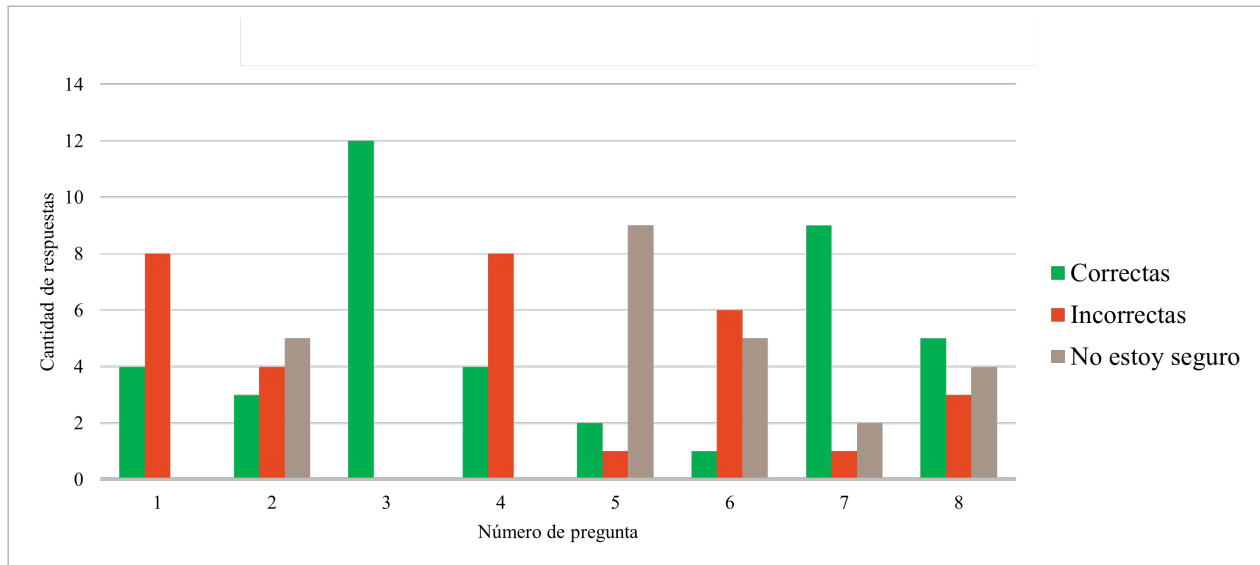
Con base en la retroalimentación obtenida durante las pruebas preliminares, el cuestionario incluyó en cada una de las preguntas la opción de “no estoy seguro”, para aquellos casos en los cuales los participantes desconocieran la respuesta. A su vez, se agregó una descripción dentro del cuestionario en la se indicaba que no debían buscar la respuesta a las preguntas, sino contestar con base en sus conocimientos actuales sobre el tema, para así poder establecer una línea base de conocimientos.

Como se observa en la figura [35](#), la mayoría de las preguntas fueron respondidas de manera errónea o, por su parte, los participantes marcaron que desconocían la respuesta a la pregunta. Si se le asigna un punto a cada pregunta respondida correctamente, se obtiene que como promedio los participantes obtuvieron una nota de 3.33. Esto indica que, en promedio y con respecto al cuestionario, los participantes únicamente conocen un 41.6 % de ciberseguridad.

Algo interesante a notar en la gráfica [35](#) es el hecho de que existen dos preguntas las cuales fueron respondidas en su mayoría de forma correcta (3 y 7). En este caso, ambas preguntas aludían a una acción que el usuario debía tomar; por ejemplo, para la pregunta 3 el usuario tenía que decidir sobre qué hacer si el banco informaba de un acceso no autorizado a su cuenta de banco y solicitaba información personal para bloquearlo. En este caso, todos los participantes indicaron que llamarían al banco para confirmar si la información es verídica. Esto es interesante debido a que, si en la práctica sucede, sería una protección contra ataques de *phishing*.

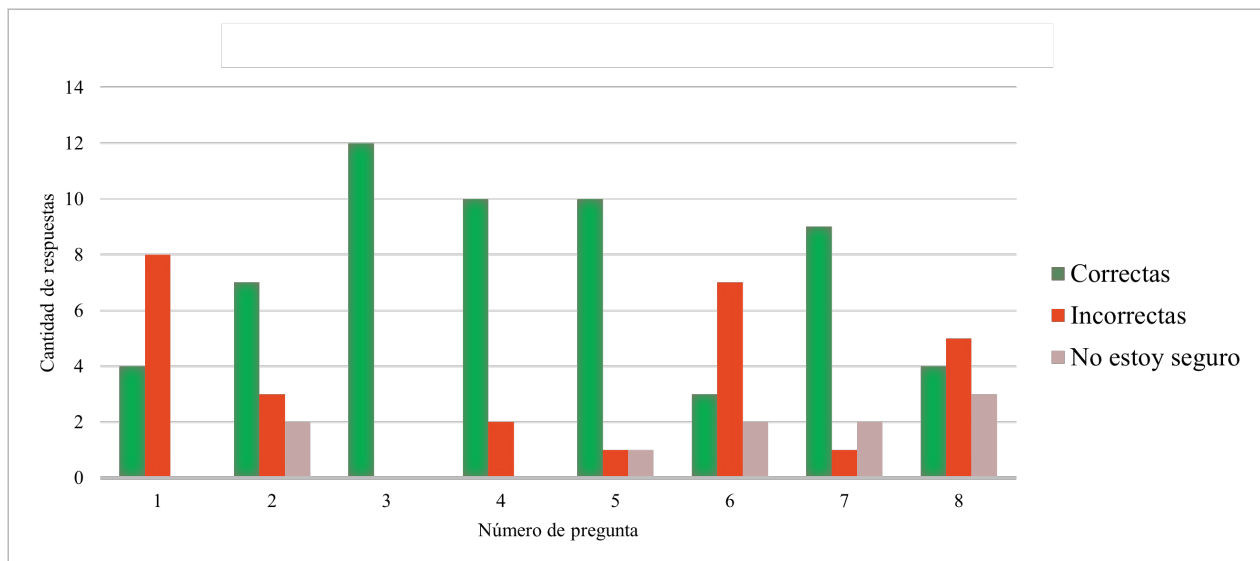
Al finalizar la prueba, para establecer el conocimiento adquirido por los participantes al usar ELA, se les compartió nuevamente el cuestionario. A simple vista, si se comparan las gráficas [35](#) y [36](#), se puede notar que a manera general existió un incremento en el número de preguntas respondidas

Figura 35. Gráfica de respuestas al cuestionario de ciberseguridad antes de usar ELA en prueba piloto



de manera correcta, ya que en la gráfica 36 estas representan un 61.45% de todas las preguntas respondidas. A su vez, se puede observar que existió una disminución en la cantidad de respuestas de “no estoy seguro”; en algunos casos esto representó un incremento en las cantidad de respuestas incorrectas (tal como en la pregunta 6), pero en su mayoría esto indicó que los participantes ya conocían la respuesta correcta.

Figura 36. Gráfica de respuestas de ciberseguridad después de usar ELA



Al hablar con los participantes sobre el incremento en su conocimiento sobre ciberseguridad,

estos mencionaron que una de las formas más fáciles mediante la cual aprendieron fue gracias a los *tips* diarios, ya que regularmente les aparecía uno en la mañana y otro en la tarde, que les daba definiciones sobre tipos de ataque (como *vishing* o *ransomware*).

8.5. Simulación de ataque de *phishing*

Luego de enviar a los usuarios el cuestionario de conocimientos, se estuvo trabajando en los ataques de *phishing* para que estos fueran lo más real posible. Una vez se tuvo todo listo, se realizó el ataque; para ello se creó un dominio el cual fue asociado a una instancia que contenía los HTML desarrollados para *phishing*.

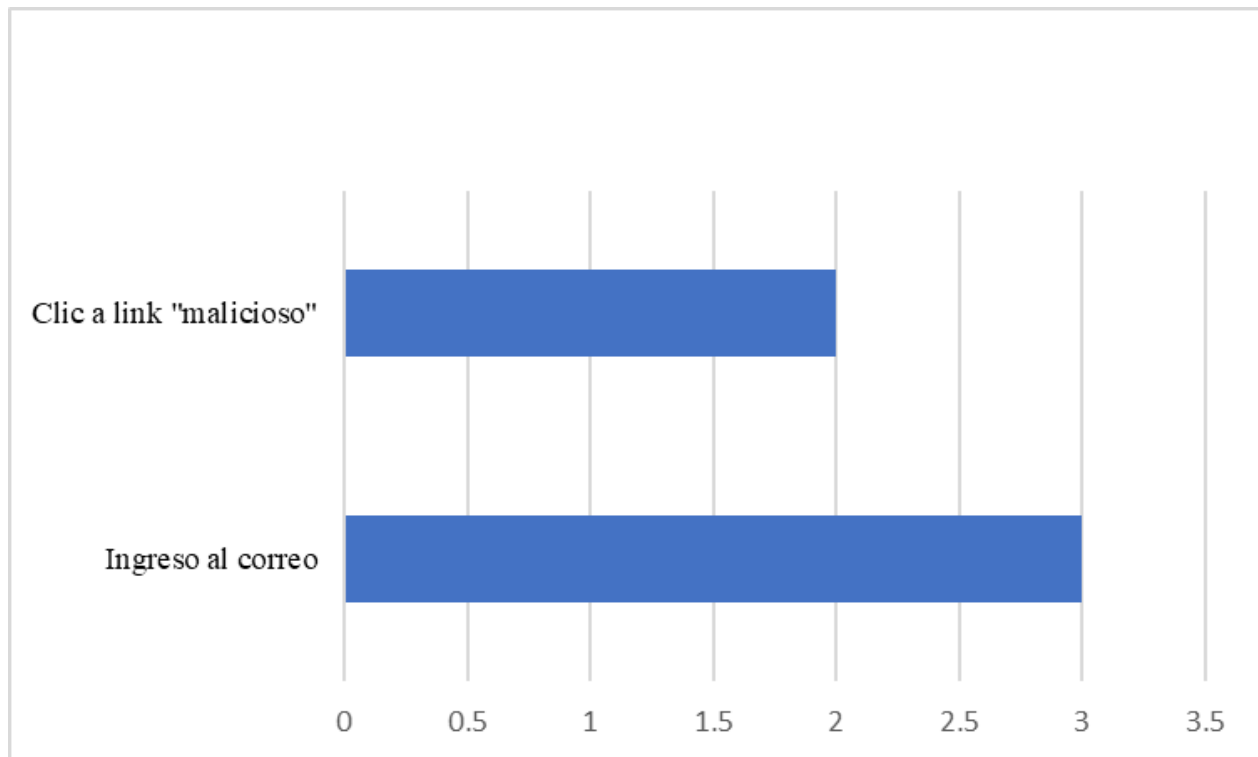
Posteriormente, debido a que se buscaba que el asistente detectara el dominio como *phishing*, se agregaron caracteres como números, símbolos y letras al dominio y se aseguró que el asistente era capaz de bloquear el dominio. Cabe resaltar que para poder tener un registro de quiénes daban clic al *link* de *phishing*, se colocaron identificadores codificados en base 64 en la URL provista en el correo o el mensaje de texto.

Uno de los retos más interesantes al momento de lanzar el ataque de *phishing* fue el hecho de que los servicios de correo electrónico detectaban como *spam* el correo enviado. Esto dificultaba que los usuarios fueran capaces de ver este correo; por ello, en algunos casos se les pidió a los participantes que revisaran la carpeta de *spam* por algún correo de una nueva actualización de ELA. Todo esto se hizo con el objeto de que el correo de *phishing* fuera percatado por el participante, pero sin proveer demasiada información que los pusiera alerta.

8.5.1. Prueba preliminar

En la prueba preliminar únicamente se lanzó un ataque de *phishing* el cual fue un correo electrónico informando sobre un cambio en la contraseña de Facebook. Los resultados obtenidos mediante la simulación de *phishing* se pueden observar en la figura [37](#), la cual indica que de los 4 participantes que recibieron el correo, 3 abrieron el correo y dos de estos le dieron clic al *link* provisto dentro del correo; sin embargo, ninguno ingresó sus credenciales en el sitio *web* falso.

Figura 37. Gráfico de interacción de usuarios con el correo de simulación de phishing



Posteriormente se tuvo un acercamiento con los usuarios para poder conocer sobre su experiencia ante este ataque simulado. La persona que únicamente abrió el correo con anterioridad ya había realizado un cambio de contraseña en Facebook, y al momento de prestar atención al remitente del correo de *phishing*, se percató de que no coincidía con los que había recibido anteriormente, por lo que ignoró dicho correo. Sin embargo, a su vez, por seguridad procedió a iniciar sesión en su cuenta de Facebook para validar que su contraseña seguía siendo la misma.

Por su parte, uno de los participantes que vio el correo y le dió clic al *link* malicioso informó que ya se había percatado de que había sido enviado como una prueba por el grupo de desarrollo de ELA y por eso navegó al *link* sin ninguna preocupación. Algo que se puede destacar en este caso es el hecho de que el asistente de seguridad le bloqueó el dominio al momento de intentar acceder.

Luego, tenemos al otro participante que también le dió clic al *link* malicioso. En este caso, la persona ya había tenido una experiencia de robo de credenciales con una de sus cuentas de redes sociales; por lo tanto, al observar el correo que indicaba que se había realizado un cambio de contraseña de su cuenta de Facebook, se asustó y leyó el correo inmediatamente. Al leer el correo, su primera impresión fue darle clic al botón que indicaba que no se había modificado la contraseña y esto hizo un redireccionamiento al inicio de sesión falso.

En el proceso de acceder al inicio de sesión, ELA le envió una notificación indicándole que el sitio al cual estaba intentando acceder probablemente era *phishing*. Esta persona anteriormente había realizado el cuestionario de conocimientos de ciberseguridad con el asistente virtual; por lo tanto, ya tenía un concepto base de lo que es *phishing*. Es por ello que inmediatamente después de la notificación del asistente procedió a cerrar la página que acababa de abrir. A su vez, inició sesión en su cuenta de Facebook para hacer un cambio de contraseña. Posteriormente, le consultó

al asistente virtual cuáles eran algunas formas de protegerse ante *phishing*, y al aprender algunas de ellas procedió a abrir nuevamente el correo para poder reconocer esos aspectos clave que pueden ayudar a prevenir que sea víctima de este tipo de ataque.

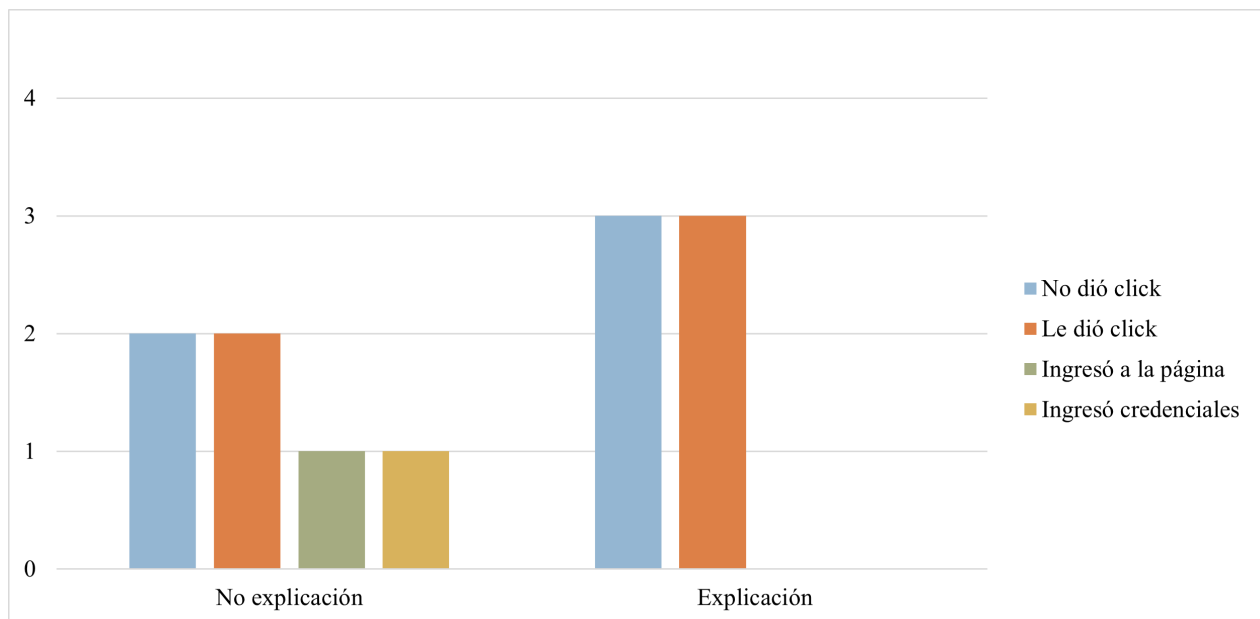
8.5.2. Prueba piloto

En el caso de la prueba piloto, se estuvieron enviando dos ataques simulados de *phishing*. El primero, mediante correo electrónico, hacía alusión sobre un error en el pago de Spotify. Y el segundo, mediante mensaje de texto, compartía un código de verificación para un cambio de contraseña de Facebook.

En este caso del correo de Spotify no se tuvo ningún acceso al sitio de *phishing*. En algunos casos esto ocurrió debido a que el correo entraba en la bandeja de *spam*; a su vez, algunos de los comentarios de los participantes con respecto a esta prueba fue el hecho de que, si bien utilizaban Spotify, en general no tenían una suscripción *premium*; por lo tanto, al momento de recibir el correo, lo único que hacían era ignorarlo. A su vez, otro comentario con respecto a este ataque fue el hecho de que dentro del mensaje enviado por correo faltaba una indicación más clara de que fuera de Spotify, ya que si bien se mantenía el estilo gráfico, el remitente no hacía alusión a Spotify y de igual forma no se tenía el *logo* dentro del correo.

Por su parte, en el caso del ataque simulado mediante mensaje de texto, las respuestas fueron mucho más rápidas que con respecto al correo electrónico, y en este caso sí se tuvo interacción con el *link* que se compartió dentro del mensaje.

Figura 38. Gráfico de interacción con el ataque de *phishing* mediante mensaje de texto



Como se observa en la figura [38](#), un 42% de los participantes ignoraron el mensaje de texto o no le dieron clic al *link* dentro del mensaje. En algunos casos esto fue a causa de que el mensaje nunca

les llegó, o por su parte, a los participantes les pareció demasiado extraño el *link* que se adjuntaba dentro del mensaje, ya que no tenía ninguna palabra relacionada con Facebook.

A su vez, dentro de las personas que sí le dieron clic al *link* dentro del mensaje, se observa que un 29% de estas sí accedió al inicio de sesión falso de Facebook, y de estos un 50% ingresó sus credenciales. Cabe resaltar que a todas las personas que le dieron clic al *link*, ELA les envió una notificación donde bloqueaba el dominio al cual estaban intentando acceder. Para quienes tenían la versión con explicación, esta notificación incluía el porqué se estaba bloqueando el dominio, y para quienes no tenían explicación, únicamente informaba del bloqueo.

Ambas notificaciones incluían la opción de permitir la navegación; sin embargo, algo interesante que informaron los participantes que tenían la explicación y le habían dado clic, fue el hecho de que debido a la explicación, en algunos casos ni se percataron de la opción de permitir la navegación. Esto, a diferencia de los participantes con la versión que no tenía explicación, evitó que algunos accedieran al sitio *web*.

Por su parte, aquellas personas que no tenían explicación y accedieron al sitio *web* comentaron que al momento de que ELA les envió la notificación, al ver la opción de permitir, una persona por curiosidad continuó navegando hasta el sitio *web* y la otra persona sí se encontraba preocupada por el posible robo de su contraseña de Facebook; por lo tanto, decidió permitir la navegación.

En el cuadro 4 se puede observar la relación de que una persona acceda a un sitio malicioso con respecto a la versión de ELA utilizada. Se puede notar que de las 4 personas que le dieron clic al *link* con la versión sin explicación, únicamente el 50% de estas se protegió, es decir, no continuó navegando a la página *web*. Por su parte, también se observa que de las 3 personas que le dieron clic al *link* de Facebook en el mensaje de *smishing* enviado, ninguna de estas continuó navegando hasta el sitio *web*, dando así una “protección” del 100%. Esto permite inferir que la explicación sobre el posible ataque del cual se puede llegar a ser víctima sí influye en la decisión del usuario de continuar o no la navegación.

Cuadro 4. *Relación de la protección de los dos tipos de versiones de ELA con el acceso al link de Facebook en ataque de smishing*

Versión	Clic a link	Acceso a página	Credenciales	Protección
No explicación	4	1	1	50%
Explicación	3	0	0	100%

A lo largo de este proyecto se desarrolló un asistente virtual mediante *large language models* capaz de responder dudas sobre ciberseguridad, de una forma clara con lenguaje amigable, mediante el proceso de *fine tuning*. Este asistente no solo provee información acerca de qué es un virus y cómo protegerse, sino a su vez, tiene una serie de *tips* y datos interesantes de ciberseguridad que buscan que, de manera rápida, un usuario pueda aprender sobre tipos de ataque y virus.

A su vez, se desarrolló un modelo de detección de dominios, el cual, mediante ciertas características léxicas (como la cantidad de números o letras en el dominio) y características estadísticas (como el tiempo activo del dominio), es capaz de clasificar en un 70 % de las veces un dominio de manera correcta; y al momento de predecir un dominio como benigno se cuenta con una precisión del 88 %. En particular, el modelo no solo realiza una clasificación binaria (si el dominio es malicioso o no), sino que también es capaz de distinguir entre distintos tipos de dominios maliciosos. Este modelo, en conjunto con el asistente virtual desarrollado, se integraron no solo en una aplicación, sino a su vez entre sí, ya que al momento de que el modelo de dominios clasifica un dominio como malicioso, el asistente virtual provee información sobre este tipo de ataque y diferentes formas de protegerse.

En lo que respecta a la capacidad del asistente para ser una herramienta educativa en ciberseguridad, se puede concluir, a partir de los resultados obtenidos en las pruebas de conocimiento, que el uso del asistente generó un incremento del 19 % en el conocimiento de los participantes. En otras palabras, tras utilizar la aplicación que incorporó el asistente virtual, se observó un aumento en la cantidad de respuestas correctas proporcionadas por los participantes.

Finalmente, tras llevar a cabo simulaciones de ataques de *phishing* y analizar las respuestas de los participantes, se puede deducir que proporcionar detalles sobre el *malware* utilizado por un dominio malicioso contribuye significativamente a aumentar la probabilidad en un 50 % de que una persona se proteja de manera exitosa ante un ataque, esto en comparación con no brindar información extra a los usuarios sobre el tipo de *malware* utilizado por un dominio.

- Para futuros desarrolladores que busquen continuar con la elaboración del proyecto, se recomienda integrar en el asistente virtual un método de retroalimentación de respuestas mediante el cual los usuarios de la aplicación sean capaces de calificar las respuestas provistas por el asistente.
- Para futuras implementaciones se recomienda utilizar la estrategia de *few shot* para el proceso de especialización del modelo, en lugar de *one shot*, para lograr que el modelo sea capaz de generalizar una entrada particular con diferentes ejemplos.
- Para futuras implementaciones se recomienda desarrollar un modelo de detección de dominios que incluya dentro de las características de entrenamiento un análisis de los n-gramas presentes en el dominio. Esto con el objetivo de que el método de clasificación analice a detalle los errores tipográficos y gramaticales presentes en un dominio.
- Para futuras implementaciones se recomienda realizar dos modelos de clasificación: el primero, que permita analizar el dominio y realizar únicamente una clasificación binaria (si el dominio es malicioso o no); y el segundo, que permita hacer una clasificación de tipo de *malware*, analizando no únicamente el dominio sino la URL completa. Esto permitirá obtener características particulares más detalladas de cada tipo de *malware* existente, buscando así mejorar la precisión de la clasificación.

-
- [1] A. G. Johansen, *What is antivirus software? Antivirus definition*, Norton, 2023. dirección: <https://us.norton.com/blog/malware/what-is-antivirus>.
- [2] Norton, *115 cybersecurity statistics + trends to know in 2023*, 2023. dirección: <https://us.norton.com/blog/emerging-threats/cybersecurity-statistics>.
- [3] H. B. Essel, D. Vlachopoulos, A. Tachie-Menson, E. E. Johnson y P. K. Baah, “The impact of a virtual teaching assistant (chatbot) on students’ learning in Ghanaian higher education,” *International Journal of Educational Technology in Higher Education*, vol. 19, n.º 1, 2022. DOI: [10.1186/s41239-022-00362-6](https://doi.org/10.1186/s41239-022-00362-6).
- [4] A. Soofastaei, “Virtual assistants,” en *Virtual Assistant*, IntechOpen, oct. de 2021. DOI: [10.5772/intechopen.100248](https://doi.org/10.5772/intechopen.100248). dirección: <https://doi.org/10.5772/intechopen.100248>.
- [5] R. Pieraccini, “What is a virtual assistant,” en *AI Assistant*, MIT Press, 2021, págs. 7-9.
- [6] T. Hope, *10 types of virtual assistants*, sep. de 2022. dirección: <https://medium.com/@ossisto.digital/10-types-of-virtual-assistants-9a4cf70a37ca>.
- [7] OpenAI, *Introducing ChatGPT*, 2022. dirección: <https://openai.com/blog/chatgpt>.
- [8] F. Clarizia, F. Colace, M. Lombardi, F. Pascale y D. Santaniello, “Chatbot: An education support system for student,” *International Journal of Educational Technology in Higher Education*, vol. 15, n.º 1, 2018. DOI: [10.1007/978-3-030-01689-0_23](https://doi.org/10.1007/978-3-030-01689-0_23).
- [9] M. V. Wüthrich y M. Merz, “Natural language processing,” en *Statistical Foundations of Actuarial Learning and its Applications*, Springer International Publishing AG, 2021, págs. 425-452. DOI: [10.1007/978-3-031-12409-9_10](https://doi.org/10.1007/978-3-031-12409-9_10).
- [10] T. Contributor, *Natural language understanding (NLU)*, Enterprise AI, 2023. dirección: <https://www.techtarget.com/searchenterpriseai/definition/natural-language-understanding-nlu>.
- [11] E. Adamopoulou y L. Moussiades, “An overview of chatbot technology,” en *IFIP Advances in Information and Communication Technology*, 2019, págs. 373-383. DOI: [10.1007/978-3-030-49186-4_31](https://doi.org/10.1007/978-3-030-49186-4_31).

- [12] S. Yolchuyeva, G. Németh y B. Gyires-Toth, “Self-attention networks for intent detection,” en *Proceedings - Natural Language Processing in a Deep Learning World*, Shoumen, Bulgaria: Incoma Ltd., oct. de 2019. DOI: [10.26615/2F978-954-452-056-4_157](https://doi.org/10.26615/2F978-954-452-056-4_157), dirección: https://doi.org/10.26615/2F978-954-452-056-4_157.
- [13] R. Karim, *Illustrated: Self-attention*, mar. de 2023. dirección: <https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a>.
- [14] X. Huang, *Chatbot: Design, architecture, and applications*, University of Pennsylvania, 2021. dirección: <https://www.semanticscholar.org/paper/CHATBOT%3A-DESIGN%2C-ARCHITECTURE%2C-AND-APPLICATIONS-Huang-Marcus/5927294d4d011f7b1e5dadbd275340d15c3e5d26e>.
- [15] C. Greyling, *Dialog management considerations for chatbots*, dic. de 2021. dirección: <https://cobusgreyling.medium.com/dialog-management-considerations-for-chatbots-6ed4dca65a80>.
- [16] Sciforce, *A comprehensive guide to natural language generation*, feb. de 2022. dirección: <https://medium.com/sciforce/a-comprehensive-guide-to-natural-language-generation-dd63a4b6e548>.
- [17] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” en *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio et al., eds., vol. 30, Curran Associates, Inc., 2017.
- [18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei e I. Sutskever, *Language models are unsupervised multitask learners*, 2019. dirección: <https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573>.
- [19] M. A. Boden, *Artificial intelligence*. Elsevier, jun. de 1996.
- [20] D. Partridge, *Artificial intelligence*. Elsevier, ene. de 2002, págs. 237-250. DOI: [10.1016/b0-12-227210-2/00038-8](https://doi.org/10.1016/b0-12-227210-2/00038-8).
- [21] Cambridge Dictionary, *Intelligence*, 2023. dirección: <https://dictionary.cambridge.org/dictionary/english/intelligence>.
- [22] S. Shaari y M. E. E. M. Matore, “Emphasizing the concept of spiritual intelligence from Islamic and Western perspectives on multiple intelligence,” *Creative Education*, vol. 10, n.º 12, págs. 2815-2830, ene. de 2019. DOI: [10.4236/ce.2019.1012208](https://doi.org/10.4236/ce.2019.1012208).
- [23] M. H. Huang y R. T. Rust, “Artificial intelligence in service,” *Journal of Service Research*, vol. 21, n.º 2, págs. 155-172, feb. de 2018. DOI: [10.1177/1094670517752459](https://doi.org/10.1177/1094670517752459).
- [24] E. Pantano y D. Scarpi, *I, robot, you, consumer: Measuring artificial intelligence types and their effect on consumers emotions in service*, 2022. dirección: <https://journals.sagepub.com/doi/full/10.1177/10946705221103538>.
- [25] B. Müller, J. Reinhardt y M. T. Strickland, *Neural networks*. Springer Science Business Media, dic. de 2012.
- [26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., “Generative adversarial nets,” *Communications of the ACM*, 2014, Original paper. dirección: <https://doi.org/10.1145/3422622>.
- [27] L. Hardesty, *Explained: Neural networks*, MIT News, abr. de 2017. dirección: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>.
- [28] D. Yeung, I. Cloete, D. Shi y W. W. Y. Ng, “Introduction to neural networks,” en *Sensitivity Analysis for Neural Networks*, Springer, 2009, págs. 1-15. DOI: [10.1007/978-3-642-02532-7_1](https://doi.org/10.1007/978-3-642-02532-7_1).

- [29] National Institute of Neurological Disorders and Stroke, *Brain basics: The life and death of a neuron*, 2023. dirección: <https://www.ninds.nih.gov/health-information/public-education/brain-basics/brain-basics-life-and-death-neuron>.
- [30] Stanford University, *Neural networks*, n.d. dirección: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Biology/index.html>.
- [31] University of California - San Diego, *Why are neuron axons long and spindly? Study shows they're optimizing signaling efficiency*, jul. de 2018. dirección: <https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html>.
- [32] R. García, "El perceptrón: Una red neuronal para clasificar datos," *Investigación en Modelos Matemáticos Aplicados a la Gestión y la Economía*, vol. 1, pág. 3, mar. de 2022.
- [33] J. M. Alvarez, *El perceptrón como neurona artificial*, jun. de 2018. dirección: <http://blog.josemarianoalvarez.com/2018/06/10/el-perceptron-como-neurona-artificial/>.
- [34] I. Goodfellow, Y. Bengio y A. Courville, *Deep learning*. MIT Press, nov. de 2016.
- [35] A. Graves, *Supervised sequence labelling with recurrent neural networks*. Springer, feb. de 2012.
- [36] A. Radford, K. Nasimhan, T. Salimans e I. Sutskever, "Improving language understanding with unsupervised learning," *OpenAI Blog*, 2018.
- [37] Q. Zhu y J. Luo, "Generative pre-trained transformer for design concept generation: An exploration," *Proceedings of the Design Society*, vol. 2, págs. 1825-1834, mayo de 2022. DOI: [10.1017/pds.2022.185](https://doi.org/10.1017/pds.2022.185).
- [38] T. B. Brown, B. Mann, N. Ryder et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, mayo de 2020. dirección: <https://arxiv.org/pdf/2005.14165.pdf>.
- [39] Cambridge Dictionary, *Security*, 2023. dirección: <https://dictionary.cambridge.org/dictionary/english/security>.
- [40] Cambridge Dictionary, *Information*, 2023. dirección: <https://dictionary.cambridge.org/dictionary/english/information>.
- [41] M. E. Whitman y H. J. Mattord, *Management of information security*, 2.^a ed. Cengage Learning, 2008.
- [42] M. E. Whitman y H. J. Mattord, *Principles of information security*, 4.^a ed. Cengage Learning, 2012.
- [43] D. Craigen, N. Diakun-Thibault y R. Purse, "Defining cybersecurity," *Technology Innovation Management Review*, vol. 4, n.º 10, págs. 13-21, 2014. DOI: [10.22215/timreview/835](https://doi.org/10.22215/timreview/835).
- [44] F. Dinha, *The human factor in cybersecurity: Understanding social engineering*, abr. de 2023. dirección: <https://www.forbes.com/sites/forbestechcouncil/2023/04/10/the-human-factor-in-cybersecurity-understanding-social-engineering/>.
- [45] P. Singer y A. H. Friedman, *Cybersecurity and cyberwar*. Oxford University Press, 2014.
- [46] C. Hadnagy, *Social engineering: The science of human hacking*. John Wiley & Sons, 2018.
- [47] Cambridge Dictionary, *Psychology*, 2023. dirección: <https://dictionary.cambridge.org/dictionary/english/psychology>.
- [48] P. O. Gray y D. Bjorklund, *Psychology*, 8.^a ed. Worth Publishers, 2018.
- [49] B. K. Wiederhold, "The role of psychology in enhancing cybersecurity," *Cyberpsychology, Behavior, and Social Networking*, vol. 17, n.º 3, págs. 131-132, 2014.

- [50] A. Torres, *Cognición: definición, procesos principales y funcionamiento*, oct. de 2022. dirección: <https://psicologiyamente.com/inteligencia/cognicion-definicion-procesos>.
- [51] K. Cherry, *Cognition in psychology*, abr. de 2023. dirección: <https://www.verywellmind.com/what-is-cognition-2794982>.
- [52] T. Kvaran, S. Nichols y A. G. Sanfey, “The effect of analytic and experiential modes of thought on moral judgment,” en *Progress in Brain Research*, Elsevier, 2013, págs. 187-196. DOI: [10.1016/b978-0-444-62604-2.00011-3](https://doi.org/10.1016/b978-0-444-62604-2.00011-3).
- [53] D. Kahneman, *Thinking, fast and slow*. Farrar, Straus y Giroux, 2011.
- [54] A. Groenewegen, *Kahneman fast and slow thinking explained*, abr. de 2022. dirección: <https://suebehaviouraldesign.com/kahneman-fast-slow-thinking/>.
- [55] D. Pucicov, *Do cognitive biases have an impact on cyber security?* 2020. dirección: <https://manazerskainformatika.sk/do-cognitive-biases-have-an-impact-on-cyber-security/>.
- [56] Santander, *The ostrich effect: when we hide from our finances*, 2022. dirección: <https://www.santander.com/en/stories/the-ostrich-effect-when-we-hide-from-our-finances>.
- [57] The Decision Lab, *Optimism bias*, n.d. dirección: <https://thedeclarationlab.com/biases/optimism-bias>.
- [58] O. Lystrup, *Cisco Umbrella releases free top 1 million sites list*, mayo de 2018. dirección: <https://medium.com/cisco-shifted/cisco-umbrella-releases-free-top-1-million-sites-list-8497fba58efe>.
- [59] A. R., *How does BERT NLP optimization model work?* Turing, nov. de 2022. dirección: <https://www.turing.com/kb/how-bert-nlp-optimization-model-works>.
- [60] Meta AI, *Introducing LLaMA: A foundational, 65-billion-parameter large language model*, feb. de 2023. dirección: <https://ai.meta.com/blog/large-language-model-llama-meta-ai/>.
- [61] Technology Innovation Institute, *Falcon*, 2023. dirección: <https://falconllm.tii.ae/falcon.html>.
- [62] M. Mitchell y D. C. Krakauer, “The debate over understanding in AI’s large language models,” *Proceedings of the National Academy of Sciences*, vol. 120, n.º 13, mar. de 2023. DOI: [10.1073/pnas.2215907120](https://doi.org/10.1073/pnas.2215907120). dirección: <https://doi.org/10.1073/pnas.2215907120>.
- [63] Cambridge Dictionary, *Fine-tuning*, sep. de 2023. dirección: <https://dictionary.cambridge.org/dictionary/english/fine-tuning>.
- [64] OpenAI, *Fine-tuning guide*. dirección: <https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset>.
- [65] Cambridge Dictionary, *Obfuscate*, 2023. dirección: <https://dictionary.cambridge.org/dictionary/english/obfuscate>.
- [66] Kaspersky, *Typosquatting: definición y explicación*, 2023. dirección: <https://latam.kaspersky.com/resource-center/definitions/what-is-typosquatting>.
- [67] J. Brownlee, *A gentle introduction to imbalanced classification*, ene. de 2020. dirección: <https://machinelearningmastery.com/what-is-imbalanced-classification/>.
- [68] D. Hara, K. Sakurai e Y. Musashi, “Classification of malicious domains by their LIFETIME,” en *Lecture Notes on Data Engineering and Communications Technologies*, Springer, ene. de 2020. DOI: [10.1007/978-3-030-39746-3_35](https://doi.org/10.1007/978-3-030-39746-3_35).

- [69] Palo Alto Networks, *What are malicious newly registered domains?* n.d. dirección: <https://www.paloaltonetworks.com/cyberpedia/what-are-malicious-newly-registered-domains>.
- [70] P. A. Méndez Morales, *Implementación de una aplicación Android para educar sobre ciberseguridad, interceptación de tráfico de red malicioso e identificación de malware basado en los permisos solicitados por las aplicaciones*, Tesis de Licenciatura, 2023.
- [71] SpamHaus, *URLhaus - Malware URL exchange*, n.d. dirección: <https://urlhaus.abuse.ch/>.

12.1. Datos de modelo de clasificación de dominios

Para el entrenamiento y validación del modelo de clasificación de dominios se utilizó el siguiente conjunto de datos:

- Conjunto de datos de *phishing*
- Conjunto de datos de *ransomware* y *malware*
- Conjunto de datos benignos

12.2. Consentimiento informado

[Acceso a consentimiento informado](#)

Figura 39. Consentimiento informado

Participación en uso de Asistente Virtual orientado a ciberseguridad que brinda un acompañamiento en la navegación de sitios web o descarga de aplicaciones de forma segura

Fecha: ___/10/2023

Estimado participante, para ser parte de este proyecto de investigación en el estudio de la aplicación de la Inteligencia Artificial en la Ciberseguridad es necesario que usted comprenda los siguientes lineamientos. Este estudio forma parte del proyecto de graduación de los estudiantes Diana Zaray Corado López y Pablo Alejandro Méndez Morales para optar al grado académico de Licenciatura en Ingeniería en Computación y Tecnologías de la Información de la Universidad del Valle de Guatemala. Este proyecto será supervisado por MSc. Douglas Barrios.

PROYECTO

En este proyecto, se le solicitará que instale y utilice una aplicación desarrollada por el grupo de trabajo. Esta aplicación es una asistente virtual llamada Ela. Ela brinda un servicio de seguridad para la navegación en línea y descarga de aplicaciones. Esto lo hace mediante el análisis de dominios y los permisos solicitados por las aplicaciones instaladas. Esto implica que la aplicación es capaz de interceptar aspectos de su navegación por internet y analizar sus aplicaciones descargadas. Esta información será recolectada únicamente para fines académicos, sin divulgar información que permita la identificación del usuario.

La duración aproximada de la prueba es de 3 días. Durante estos días se estará compartiendo una serie de pruebas sensibles en un ambiente controlado. De manera que, no se recolectará información sensible y la información recopilada por estas pruebas mantendrá el anonimato. El objetivo principal de este experimento es que pueda interactuar con la aplicación al activar la protección mediante Ela y detallar su experiencia diaria.

RIESGOS Y BENEFICIOS

Se debe de tomar en consideración que la aplicación está en versión beta, por lo tal, su uso en sí representa un riesgo para los usuarios. Los modelos de inteligencia artificial, tanto el de detección de aplicaciones maliciosas y de dominios maliciosos, fueron entrenados sobre una serie de patrones particulares que buscan generalizarse. Por lo tal, es posible que clasifiquen un dominio o aplicación erróneamente, implicando que se bloquee el acceso a un sitio benigno o que se permita el acceso a un sitio malicioso.

Entre los beneficios por participar en este estudio, se espera que se adquiera conocimiento acerca de ciberseguridad y brindar apoyo en el desarrollo de una herramienta que busca reducir el alto índice de ataques cibernéticos.

Consentimiento

Yo, _____ autorizo y estoy de acuerdo con que se utilice la información recopilada para fines académicos. Mi participación es voluntaria y exonero de responsabilidad al grupo de trabajo y a la Universidad Del Valle de Guatemala. Asimismo, autorizo que se registre mi participación y aparición en fotografías o futuras publicaciones por parte de la UVG o el grupo de trabajo mencionado anteriormente. Doy autorización para distribuir, mostrar públicamente de forma física o electrónica la información dada durante esta prueba.

Firma del participante

12.3. Formulario de pruebas para evaluación de aceptación de modelos LLM

Pruebas de prototipo

Basado en el uso del asistente virtual en la página web, responde las siguientes preguntas

1. Cuando querías saber información o definición sobre algún tipo de virus o ataque informático. ¿Qué respuesta le gustó más?



Marca solo un óvalo.

- Respuesta rápida
 Respuesta lenta

2. Cuando querías algún tip de cómo protegerse ante un tipo de ataque cibernético. ¿Qué tipo de respuesta le pareció más útil?



Marca solo un óvalo.

- Respuesta corta
 Respuesta larga

3. Cuando querías formas de cómo detectar algún tipo de ataque. ¿Qué respuesta le gustó más?

Marca solo un óvalo.

- Respuesta corta
 Respuesta larga

4. Describe su experiencia en el uso del prototipo inicial del asistente virtual. ¿Qué tipo de respuesta le gustaría que le muestre el asistente cuando le haga una consulta?

12.4. Cuestionario de conocimiento en ciberseguridad de prueba piloto

Conocimiento en Ciberseguridad

Estimado participante

Somos estudiantes de la Universidad del Valle de Guatemala. El objetivo del estudio es recolectar información para realizar un análisis sobre sus conocimientos sobre ciberseguridad.

La información obtenida a través de este estudio será mantenida bajo estricta confidencialidad y su nombre no será utilizado y/o revelado. Usted tiene el derecho de retirar el consentimiento para la participación en cualquier momento. Adjunto a este formulario encontrará un consentimiento informado el cual deberá leer antes de continuar con la prueba.

No recibirá ningún tipo de compensación por participar.

¡Agradecemos su participación y el aporte brindado a nuestro trabajo de investigación!

* Indica que la pregunta es obligatoria

1. He leído el procedimiento descrito arriba y el consentimiento informado adjunto. El(la) investigador(a) me ha explicado la finalidad del estudio y en qué consiste. He contestado las preguntas de forma voluntaria y doy mi consentimiento para participar en el estudio.

Consentimiento informado

Marca solo un óvalo.

- Acepto participar en el estudio
 No acepto participar en el estudio

Nota. Parte 1 del cuestionario.

Información personal

2. ¿Cuál es tu nombre?*

3. Ingresa tu correo electrónico personal*

4. ¿Cuántos años tienes?*

5. ¿Cuál es tu sexo?*

Marca solo un óvalo.

Femenino
 Masculino
 Prefiero no decirlo

6. ¿Cuál consideras que es su nivel de conocimiento en ciberseguridad?*

Marca solo un óvalo.

1 2 3 4 5
Nada Muy en experto

7. ¿Consideras que tener conocimiento en ciberseguridad es importante?*

Marca solo un óvalo.

1 2 3 4 5
Nada Es muy importante

8. ¿Qué tanto consideras que influye el factor humano en la defensa a un ataque cibernético?*

Marca solo un óvalo.

1 2 3 4 5
Nada Influye mucho

Nota. Parte 2 del cuestionario.

Consentimientos

Esta sección es puramente para evaluar cuánto conoces de ciberseguridad actualmente. Se le solicita que responda honestamente y no busque las preguntas en internet.

9. ¿Cuál es su política de administración de contraseñas? *

Marca solo un óvalo.

- Reutilizo contraseñas
- No suelo reutilizar, pero utilizo combinaciones de contraseñas pasadas
- No reutilizo contraseñas
- Utilizo un administrador de contraseñas
- Otro: _____

10. ¿Cuáles de estas contraseñas consideras que son seguras? *

Selecciona todas las que correspondan

- P@p1to123\$, es segura porque tiene más de 8 caracteres, minúsculas, mayúsculas, números y símbolos
- C@nt@t@nd@s, es segura porque utiliza una combinación de números y letras
- J@su@t@Amo, es segura porque a nadie se le podría ocurrir
- 10011985, es segura porque son varios dígitos y al ser mi fecha de cumpleaños es fácil de recordar
- Ninguna de las contraseñas parece segura

11. A Ana le llegó un correo aparentemente inofensivo que tenía como adjunto un pdf. Al descargar el pdf, se percató de que el icono de sus archivos cambió a un candado. Luego recibió un correo indicando que toda su información había sido encriptada y necesitaba pagar Q100,000 para recuperarlos. *

Este es un ejemplo de:

Marca solo un óvalo.

- Spam
- Phishing
- Doxing
- Ransomware
- No estoy seguro

Nota. Parte 3 del cuestionario.

12. Recibiste un correo electrónico informándote que alguien accedió a tu cuenta de banco. Para poder bloquear este acceso, debes de proporcionar tus datos personales de manera urgente. *

¿Qué decides hacer?

Marca solo un óvalo.

- No tengo nada en esa cuenta de banco, por lo que ignoro el correo
- Me parece un poco raro que el banco solicite esta información, por lo cual me pongo en contacto con mi banco para verificar si existe un nuevo acceso
- Me preocupa lo que puedan hacer con mi cuenta de banco, por lo que rápidamente respondo toda la información que solicita el correo



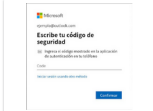

13. ¿Cuando utilizas redes sociales, cuántos detalles personales eliges revelar sobre tu vida diaria, relaciones y actividades? *

Marca solo un óvalo.

- Me encanta compartir mi día a mis amigos en las redes sociales, por algo es una red social
- Suelo compartir información siempre y cuando esta no revele datos muy personales
- Evito compartir información personal en redes sociales. Otras personas no tienen que saber ese tipo de detalles
- No uso redes sociales

14. ¿Cuál de los siguientes es un ejemplo de autenticación de dos pasos? *

Marca solo un óvalo.

	
<input type="radio"/> Opción 1	<input type="radio"/> Opción 2
	
<input type="radio"/> Opción 3	<input type="radio"/> Opción 4

Nota. Parte 4 del cuestionario.

15. ¿Qué es el vishing? *

Marca solo un óvalo.

- Un ataque de phishing que se enfoca en correos electrónicos
- Un ataque que se realiza mediante llamadas telefónicas para engañar a la víctima y obtener información confidencial
- Es un tipo de malware que se propaga sin necesidad de un programa anfitrión
- Es una técnica de cifrado de datos
- No estoy seguro

16. ¿Cuál de las siguientes afirmaciones describe mejor la ingeniería social? *

Marca solo un óvalo.

- Es una técnica para explotar vulnerabilidades en el software
- Es un campo de la ingeniería que se encarga del desarrollo de soluciones sociales
- Es una técnica psicológica que busca la obtención de información de otra persona
- No estoy seguro

17. Últimamente tu computadora está más lenta de lo habitual y crees que tienes un virus. *

¿Qué decides de hacer?

Marca solo un óvalo.

- Descargar un antivirus
- Reiniciar la computadora
- Agregar más memoria RAM
- No estoy seguro

18. Si debes hacer una cuenta en una página que no se ve segura. ¿Qué decides hacer? *

Marca solo un óvalo.

- Utilizar tu correo electrónico usual, total no tienes nada importante
- Utilizar una cuenta de correo alterna
- Usar el correo de un amigo o conocido
- No estoy seguro

19. ¿Cuál de las siguientes prácticas es esencial para realizar copias de seguridad? *

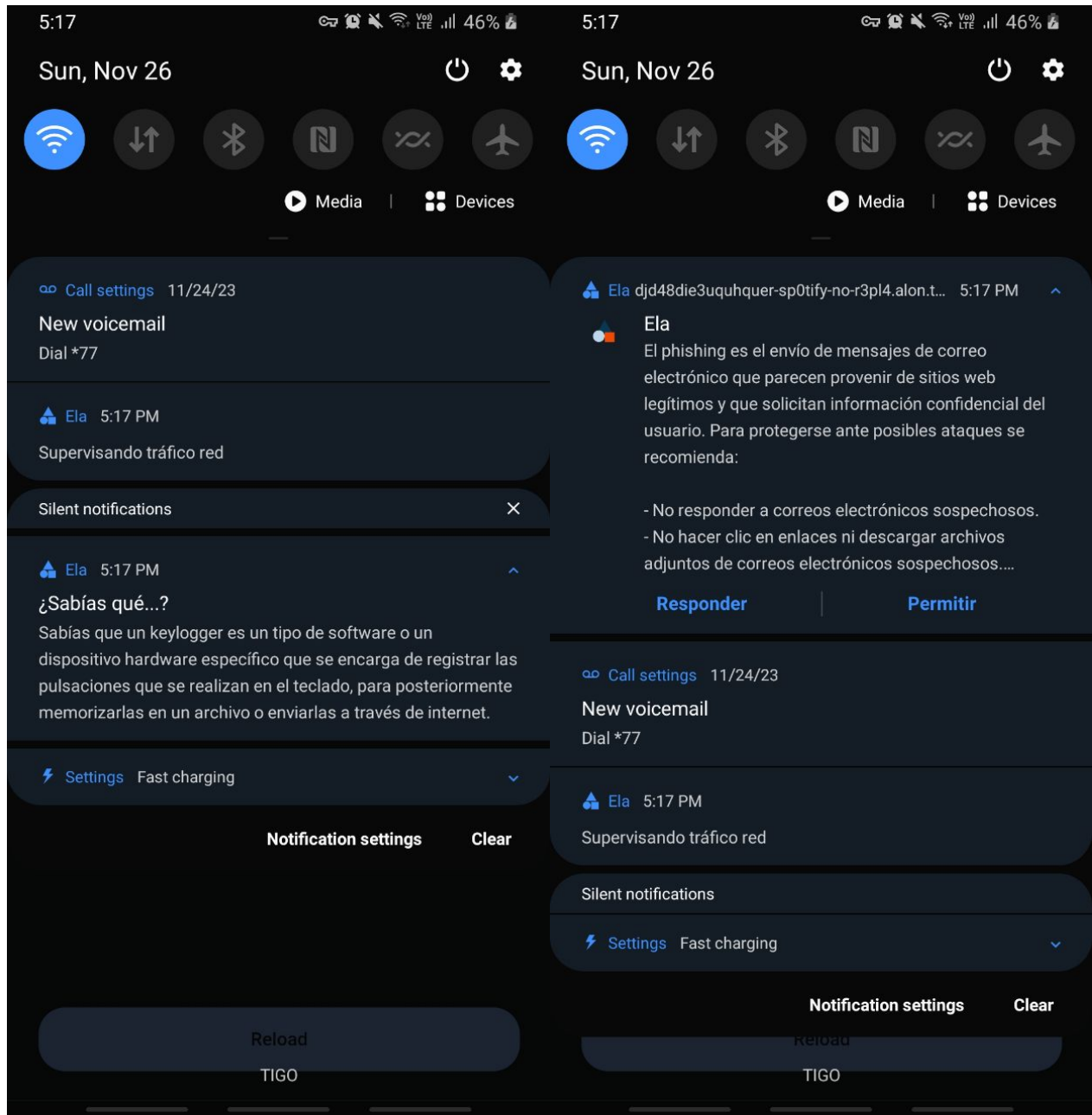
Marca solo un óvalo.

- Nunca verificar la integridad de las copias de seguridad
- Encriptar las copias de seguridad para proteger la información
- Guardar las copias de seguridad en el mismo dispositivo del que se está haciendo copia
- Ninguna de las anteriores
- No estoy seguro

Nota. Parte 5 del cuestionario.

12.5. Aplicación

Figura 40. Capturas de pantalla de la aplicación ELA



(a) Notificación de tip diario

(b) Mensaje al bloquear un dominio

12.6. Test de significancia para los resultados de la simulación de ataque de *phishing*

Dado que se están evaluando variables categóricas y que las muestras son pequeñas, se utilizará la prueba exacta de Fisher para determinar si los resultados son significativamente diferentes entre los dos grupos.

12.6.1. Clics en enlaces maliciosos

Objetivo: evaluar el impacto de las notificaciones en la aplicación para la prevención de clics en enlaces maliciosos y el ingreso de credenciales.

HN: la proporción de personas que dan clic en enlaces maliciosos es igual en el grupo que recibe explicación y el que no la recibe.

HA: la proporción de personas que dan clic en enlaces maliciosos no es igual en el grupo que recibe explicación y el que no la recibe.

Cuadro 5. *Tabla de contingencia de clics en simulación de ataques de phishing*

	Sin explicación	Con explicación
Dio clic	4	3
No dio clic	2	3

Se realiza el cálculo de *p-value* y de la tasa de probabilidad mediante la librería de Python de `scipy.stats` y se obtienen los siguientes valores:

- *p-value* = 1
- Tasa de probabilidad = 2

Con un valor $p = 1$ y un valor de significancia de 0.05, no hay evidencia estadística suficiente para aceptar la hipótesis alterna de que la proporción de personas que dan clics en enlaces maliciosos no es igual entre los grupos con y sin explicación. Esto indica que, con los datos actuales, no se observa una diferencia significativa entre las personas que dan clics a enlaces maliciosos con notificaciones con y sin explicación. Sin embargo, algo que se puede resaltar es el hecho de que la tasa de probabilidad es de 2; esto básicamente indica que el grupo sin explicación tiene el doble de probabilidades de hacer clic en el enlace malicioso en comparación con el grupo con explicación. En otras palabras, la presencia de una explicación parece reducir la probabilidad de que los usuarios hagan clic en el enlace malicioso.

12.6.2. Navegación en páginas de *phishing*

HN: la explicación en las notificaciones de la aplicación no reduce la cantidad de ingresos a páginas de *phishing*.

HA: la explicación en las notificaciones de la aplicación reduce la cantidad de ingresos a páginas de *phishing*.

Cuadro 6. Tabla de contingencia para el ingreso a una página de *phishing*

	Sin explicación	Con explicación
Ingresó a página	2	0
No ingresó a página	4	6

Se realiza el cálculo de *p-value* y de la tasa de probabilidad mediante la librería de Python de `scipy.stats` y se obtienen los siguientes valores:

- $p\text{-value} = 0.455$

Con un valor $p = 0.455$, no hay evidencia estadística suficiente para rechazar la hipótesis nula de que la explicación en las notificaciones de la aplicación no reduce la cantidad de ingresos en páginas de *phishing*.

12.7. Entrenamiento de modelo de clasificación de dominios con un mayor número de datos de *ransomware*

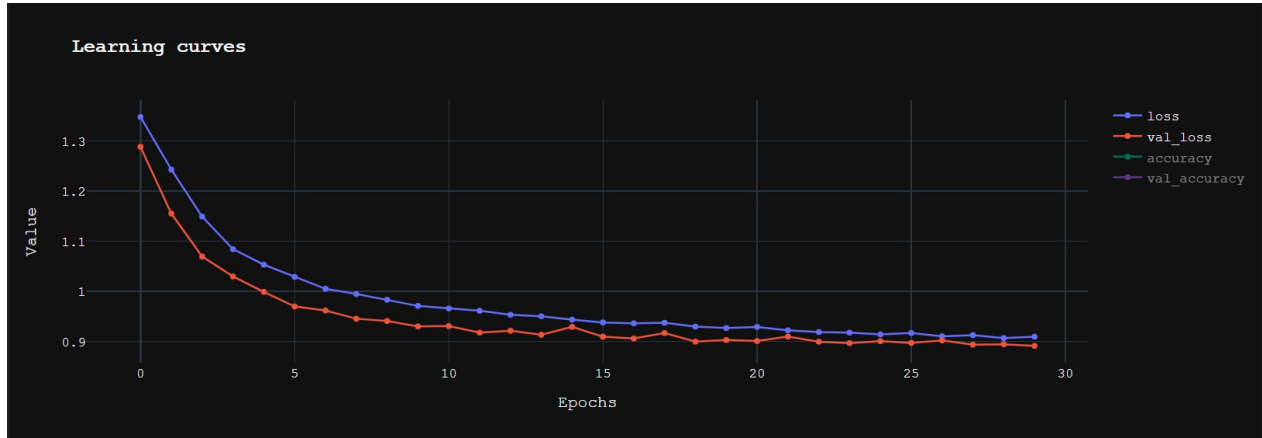
Este nuevo entrenamiento se realizó con el objeto de poder evaluar si un incremento en la cantidad de datos de dominios de *ransomware* permite mejorar la métrica de precisión de clasificación de dominio de dicho tipo.

Para esto se investigaron nuevos dominios de *ransomware*. Estos nuevos dominios se encontraron en la plataforma de URLHaus, la cual es una plataforma que se encarga de recolectar y verificar URL (*Uniform Resource Locator* o direcciones web) maliciosas con el objeto de poder ayudar a prevenir ataques maliciosos [71]. Dentro de esta plataforma se pudieron encontrar 1442 URL maliciosas de tipo *ransomware*.

Con las 1442 nuevas URL se tiene un total de 3346 datos de *ransomware* para poder entrenar. Sin embargo, se debe resaltar el hecho de que, debido a que el modelo se especializa en la clasificación de dominios y no de URL, se tuvo que realizar un proceso para eliminar dominios duplicados, dejando así 2732 dominios únicos de tipo *ransomware*. La distribución total de los datos para el desarrollo del modelo quedó de la siguiente manera:

- Benigno = 999988
- *Malware* = 435405
- *Phishing* = 98860
- *Ransomware* = 2732

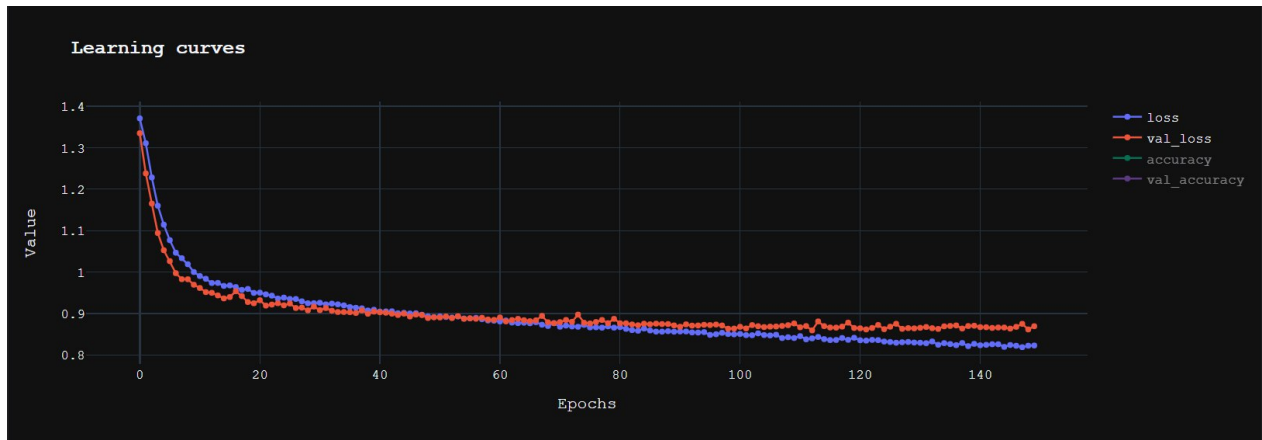
Figura 42. Curva de aprendizaje de entrenamiento de modelo de detección de dominios maliciosos con incremento en datos de ransomware y modificación en épocas de entrenamiento y tasa de aprendizaje



Estos datos comparados con la distribución del entrenamiento original incrementaron en un 43% la cantidad de datos de *ransomware*.

Utilizando el mismo modelo desarrollado y únicamente los nuevos datos de *ransomware* se obtuvieron los siguientes resultados:

Figura 41. Curva de aprendizaje de entrenamiento de modelo de detección de dominios maliciosos con incremento en datos de ransomware



La Figura 41 representa la curva de aprendizaje del modelo con los nuevos datos. Como se puede observar, luego de alrededor de 60 épocas, parece ser que el modelo se está sobreentrenando debido a que el error de los datos de validación es mayor comparado con el error de los datos de entrenamiento. Esto indica que el modelo está aprendiendo tan bien de los datos de entrenamiento que no le está siendo posible generalizar.

Para poder mejorar el modelo, se decidió reducir la cantidad de épocas de entrenamiento: de 150 se pasó a utilizar 30; esta reducción se hizo basada en las curvas de aprendizaje del modelo. A su vez, se incrementó la tasa de aprendizaje del modelo con el objeto de buscar una curva de aprendizaje ideal. De utilizar una tasa de aprendizaje de 0.00022 se pasó a utilizar una tasa de 0.00032. Con estas modificaciones se pudieron obtener las curvas de aprendizaje de la Figura 42.

Una vez se entrenó y ajustó el modelo, se realizaron algunas pruebas para evaluar las predicciones con datos que el modelo nunca ha visto. En la Figura 43 se puede observar cómo el nuevo modelo, comparado con el modelo anterior, incrementó el porcentaje de precisión para la clasificación de dominios de tipo *ransomware*.

Figura 43. Reporte de clasificación de modelo de detección de dominios maliciosos con incremento en datos de *ransomware* y modificación en épocas de entrenamiento y tasa de aprendizaje

	precision	recall	f1-score	support
Benigno	0.62	0.71	0.66	1493
Malware	0.70	0.45	0.55	1581
Phishing	0.52	0.71	0.60	1768
Ransomware	0.92	0.59	0.72	825
accuracy			0.62	5667
macro avg	0.69	0.61	0.63	5667
weighted avg	0.65	0.62	0.62	5667

Algo que se debe destacar es que este modelo también redujo el porcentaje de precisión para la clasificación de dominios benignos. Ya que pasó de tener un 88% de precisión al clasificar dominios benignos a estar en un 62%; esto es esencialmente perjudicial para el ámbito de estudio de este proyecto, ya que en el 38% de los casos el modelo está clasificando dominios maliciosos como dominios benignos, lo cual pone en peligro a los usuarios, ya que no se alerta correctamente para prevenir que sean víctimas de un ataque malicioso.