UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Implementación de una Aplicación Android para educar sobre ciberseguridad, intercepción de tráfico de red malicioso e identificación de *malware* basado en los permisos solicitados por las aplicaciones

Trabajo de graduación presentado por Pablo Alejandro Méndez Morales para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de Información

Guatemala,

2023

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

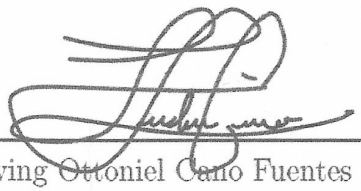
Implementación de una Aplicación Android para educar sobre ciberseguridad, intercepción de tráfico de red malicioso e identificación de *malware* basado en los permisos solicitados por las aplicaciones

Trabajo de graduación presentado por Pablo Alejandro Méndez Morales para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de Información

Guatemala,

2023

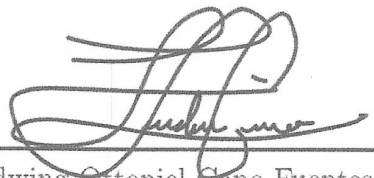
Vo.Bo.:



(f) _____

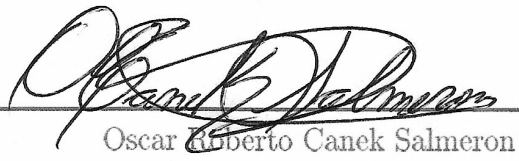
Ludwing Ottoniel Cano Fuentes

Tribunal Examinador:



(f) _____

Ludwing Ottoniel Cano Fuentes



(f) _____

Oscar Roberto Canek Salmeron

(f) _____

Douglas Leonel Barrios Gonzalez

Fecha de aprobación: Guatemala, 6 de diciembre de 2023.

Índice

Lista de figuras x

Lista de cuadros xi

Resumen xiii

Abstract xv

1. Introducción 1

2. Justificación 3

3. Objetivos 5
3.1. Objetivo general . . . 5
3.2. Objetivos específicos . . . 5

4. Alcance 7

5. Marco teórico 9
5.1. Sistemas operativos . . . 9
5.1.1. Permisos . . . 10
5.1.2. Sistemas operativos modernos . . . 10
5.2. *Hardware* . . . 11
5.2.1. CPU . . . 11
5.2.2. Memoria RAM . . . 12
5.2.3. Tarjeta madre . . . 12
5.2.4. Disco duro . . . 12
5.3. *Software* . . . 12
5.3.1. Aplicaciones móviles . . . 12
5.3.2. Lenguajes de programación . . . 13
5.3.3. API . . . 14
5.3.4. *Framework* . . . 15
5.3.5. Base de datos . . . 15

5.3.6. Paralelismo . . . 15
5.4. Redes de computadoras . . . 16
5.4.1. Tipos de redes . . . 16
5.4.2. Direcciones IP . . . 17
5.4.3. Subredes . . . 17
5.4.4. Dominios . . . 19
5.4.5. Protocolos . . . 19
5.4.6. Capas OSI . . . 20
5.4.7. *Firewall* . . . 24
5.4.8. VPN . . . 24
5.5. Ciberseguridad . . . 25
5.5.1. Amenazas . . . 25
5.5.2. Dominios maliciosos . . . 26
5.5.3. El factor humano en la ciberseguridad . . . 26
5.5.4. El estado actual de la ciberseguridad en Android . . . 26
5.6. Inteligencia Artificial . . . 27
5.6.1. Modelos de inteligencia Artificial . . . 27
5.7. Metodologías de desarrollo . . . 28
5.7.1. Scrum . . . 28
5.7.2. Kanban . . . 29

6. Metodología **31**
6.1. Pruebas de usabilidad . . . 31
6.1.1. Pruebas de corto plazo . . . 31
6.1.2. Pruebas de largo plazo . . . 31
6.2. Pruebas de aprendizaje . . . 32
6.3. Metodología de desarrollo . . . 32

7. Resultados **33**
7.1. Elección de herramientas . . . 33
7.1.1. Elección de *framework* . . . 33
7.1.2. Elección del lenguaje de programación . . . 35
7.1.3. Elección de base de datos . . . 35
7.2. Proceso de desarrollo . . . 36
7.2.1. Cronograma . . . 36
7.2.2. Sprints . . . 36
7.3. Pruebas de usabilidad . . . 40
7.3.1. Pruebas de corto plazo . . . 40
7.3.2. Pruebas de largo plazo . . . 41
7.4. Módulos . . . 42
7.4.1. Módulo de bloqueo de dominios maliciosos . . . 42
7.4.2. Módulo de aplicaciones sospechosas . . . 46
7.4.3. Módulo de asistente virtual . . . 47
7.4.4. Módulo de dato curioso de ciberseguridad . . . 47
7.5. Prueba preliminar de aprendizaje . . . 48
7.6. Prueba piloto de aprendizaje . . . 51
7.6.1. Ataque simulado . . . 52
7.6.2. Conocimiento de ciberseguridad . . . 56

8. Conclusiones **61**

9. Recomendaciones **63**

10. Bibliografía **65**

11. Anexos **69**
11.1. Capturas de pantallas de la aplicación 69
11.2. Encuestas realizadas 72

12. Glosario **89**

Lista de figuras

1. Estructura de un HTTP URL . . . 24
2. Estructura de una neurona[41] . . . 27
3. Red neuronal . . . 28
4. Árbol de decisiones sobre nadar o no nadar . . . 28

5. Cronograma . . . 36
6. Notificación al haber bloqueado un dominio malicioso . . . 43
7. Pantalla de ver detalles sobre los dominios bloqueados hoy . . . 44
8. Primer acercamiento para el bloqueo de tráfico red . . . 44
9. Segundo acercamiento para el bloqueo de tráfico red . . . 45
10. Tercer acercamiento para el bloqueo de tráfico red . . . 45
11. Pantalla de aplicaciones sospechosas . . . 46
12. Pantalla de solicitarle una pregunta a Ela . . . 47
13. Notificación del consejo diario . . . 48
14. Correo de *phishing* para la prueba preliminar . . . 50
15. Página de inicio de sesión falsa para la prueba preliminar . . . 50
16. Página mostrada cuando tratan de iniciar sesión . . . 51
17. Edades de los participantes en las pruebas de aprendizaje . . . 52
18. Sexo de los participantes en las pruebas piloto de aprendizaje . . . 52
19. Correo enviado para la prueba de aprendizaje . . . 53
20. Página de inicio de sesión falsa para la prueba de aprendizaje . . . 53
21. Resultados de las pruebas de *phishing* de *Spotify* . . . 53
22. Mensaje de texto sobre un *phishing* de Facebook . . . 54
23. Resultados de las pruebas de *phishing* . . . 56
24. Evolución de las pregunta de conocimientos de ciberseguridad sobre *malware* antes y después de haber usado Ela . . . 56
25. Evolución de las pregunta de conocimientos de ciberseguridad sobre reacciones ante el *phishing*, antes y después de haber usado Ela . . . 57
26. Evolución de las pregunta de conocimientos de ciberseguridad sobre autenticación multifactor antes y después de haber usado Ela . . . 57
27. Evolución de las pregunta de conocimientos de ciberseguridad sobre otros tipos de *malware* antes y después de haber usado Ela . . . 58

28. Evolución de las pregunta de conocimientos de ciberseguridad sobre ingeniería social antes y después de haber usado Ela 58
29. Evolución de las pregunta de conocimientos de ciberseguridad sobre reacciones ante el *malware* antes y después de haber usado Ela 59
30. Evolución de las pregunta de conocimientos de ciberseguridad sobre copias de seguridad antes y después de haber usado Ela 59
31. Evolución de las pregunta de conocimientos de ciberseguridad sobre uso de redes sociales antes y después de haber usado Ela 59
32. Evolución de las pregunta de conocimientos de ciberseguridad sobre páginas inseguras antes y después de haber usado Ela 60

33. Pantalla de inicio al tener varias aplicaciones sospechosas 69
34. Pantalla de inicio al haber bloqueado un dominio en ese día 70
35. Pantalla de ajustes 70
36. Pantalla de agregar un dominio permitido vacía 71
37. Pantalla de agregar un dominio permitido 71
38. Notificación al bloquear un dominio (versión burbuja) 72
39. Botón de acceso rápido para encender a Ela 72

Lista de cuadros

1. Recursos utilizados para renderizar una lista dinámica de 100 objetos[45] . . . 34

Resumen

Se desarrolló una aplicación para teléfonos Android que puede detectar aplicaciones sospechosas basado en los permisos que solicita cada aplicación. También puede interceptar tráfico DNS y filtrar a aquellos dominios catalogados como maliciosos; presentando una advertencia de cómo protegerse contra este tipo de ataques. La aplicación demostró ser efectiva para prevenir tráfico red malicioso y para indicarle al usuario cómo debería reaccionar bajo ese tipo de situaciones.

Abstract

An application was developed for Android phones that can detect suspicious applications based on the permissions requested by each application. It can also intercept DNS traffic, filter out those domains that were classified as malicious, and present a warning on how to protect against these types of attacks. The application proved to be effective in preventing malicious traffic and telling the user how they should react under these types of situations.

CAPÍTULO 1

Introducción

En la era actual de la digitalización, la ciberseguridad se ha convertido en un parte fundamental de la vida *online*. A pesar de que cada vez se está generando mejores técnicas para proteger a las personas de estos peligros, estas amenazas también se encuentran evolucionando constantemente.

A raíz de la situación anterior, se desarrolló el siguiente trabajo donde se diseña e implementa una aplicación para dispositivos Android. Esta aplicación permitirá bloquear tráfico red a dominios maliciosos y detectar cuáles otras aplicaciones instaladas podrían ser maliciosas. Más importante, la aplicación trata de realizar todo lo anterior educando al usuario sobre el tipo de amenazas bloqueadas y qué podría hacer para protegerse. La aplicación también responde dudas de ciberseguridad. Ayudando al desarrollo de conocimientos de ciberseguridad del usuario.

Finalmente, se pondrá a prueba si el conocimiento adquirido por el uso de la aplicación sí aumenta las probabilidades de que el usuario se proteja de otro tipo de ataques.

CAPÍTULO 2

Justificación

La violación de privacidad, daño a la reputación, suplantación de identidad y daños financieros son solo algunas de las consecuencias que se pueden obtener al ser víctima de un incidente de ciberseguridad.

Ya no es suficiente depender de tecnología para detener estas amenazas. Es necesario que este tipo de peligros puedan ser evitados desde el nivel humano. De manera que se debe conscientizar a las personas de ciberseguridad, las amenazas actuales y las acciones que puede realizar para defenderse de cada una de ellas.

Asimismo, los dispositivos móviles y otros dispositivos inteligentes generalmente cuentan con bajas defensas contra ataques cibernéticos. Esto es especialmente preocupante en dispositivos Android debido al reciente auge en la cantidad de *malware* desarollado para esa plataforma[1].

Dado todo lo anterior, se consideró necesario realizar una aplicación capaz de proteger y educar a sus usuarios al mismo tiempo.

Este proyecto propone diseñar e implementar una aplicación Android capaz de detectar y bloquear tráfico de red a dominios maliciosos; mientras se enseña al usuario sobre ciberseguridad. Finalmente, trata de evaluar si tener mejor conocimiento de ciberseguridad es suficiente para estar más seguro en el panorama *online* actual.

CAPÍTULO 3

Objetivos

3.1. Objetivo general

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

3.2. Objetivos específicos

- Interceptar tráfico red a dominios maliciosos y ser capaz de bloquearlos
- Mostrarle al usuario aplicaciones potencialmente maliciosas que tiene en su dispositivo.
- Educar sobre los diferentes tipos de ataques de ciberseguridad y cómo defenderse de ellos.

CAPÍTULO 4

Alcance

El proyecto busca mejorar la manera a la que el ser humano responde a incidentes de ciberseguridad. Parar lograr esto se utilizaron dos enfoques: conscientizar sobre ciberseguridad y proveer nuevas herramientas en este ámbito en el área móvil.

Aunque el trabajo usará algunos modelos de inteligencia artificial, esta no es una de las habilidades que se desean evaluar. Todos los modelos utilizados fueron desarrollados por terceros quienes conocen y están de acuerdo de su uso para la realización de este trabajo.

Finalmente, se espera que este trabajo evidencie el uso habilidades de utilizar la informática para llegar a soluciones que puedan ser beneficiosas para la sociedad. Específicamente, se centra en las habilidades de diseño de aplicaciones, programación de aplicaciones móviles, redes de computadoras y programación paralela.

CAPÍTULO 5

Marco teórico

5.1. Sistemas operativos

Las computadoras existen para facilitar la vida humana; comúnmente a través de aplicaciones. Asimismo, es común que esas aplicaciones requieran utilizar algún componente físico de la computadora como el disco duro, micrófono, pantalla, memoria, etc. Por si fuera poco, ahora es común que una computadora ejecute varias aplicaciones al mismo tiempo.

A raíz de todo esto nacieron los sistemas operativos. En términos generales, un sistema operativo es un programa que administra los componentes físicos de un computador. De igual forma, el sistema operativo es el encargado de darle al resto de aplicaciones el ambiente adecuado para ejecutarse correctamente [2][3].

Las tareas principales del sistema operativo son:

- Utilizar el equipo físico de la computadora de la manera más eficiente
- Ocultar la complejidad de manejar el equipo físico de la computadora
- Darle al usuario un método fácil y conveniente para interactuar con el sistema[3]

Aunque cada sistema operativo es diferente y resuelve diferentes necesidades, comúnmente proveen lo siguiente:

- Proveer un sistema de archivos
- Balancear el uso del CPU y memoria RAM entre todas las aplicaciones
- Aislar las diferentes aplicaciones
- Mostrarle al usuario un entorno gráfico donde puede interactuar con otras aplicaciones
- Proveer seguridad en contra de aplicaciones con mal comportamiento

5.1.1. Permisos

Una de las maneras principales en la que los sistemas operativos protegen en contra de aplicaciones maliciosas o incorrectamente programadas es limitando las acciones que puede realizar este programa[2].

Por ejemplo, una aplicación no puede acceder directamente a recursos del sistema, debe hacerlo a través de una solicitud al sistema operativo. Luego, el sistema operativo decidirá si le proporciona a la aplicación el permiso de usar el recurso. Entre los recursos restringidos por el sistema operativo se encuentra el sistema de archivos y el acceso a diferentes dispositivos de entrada y salida[2].

5.1.2. Sistemas operativos modernos

Debido a la variedad de necesidades humanas, existen diferentes tipos de sistemas operativos orientados a resolver distintos problemas. Aún limitándose a los sistemas operativos diseñados para computadoras de uso personal, todavía existe una cantidad de alternativas que podría abrumar a cualquiera.

No obstante, dentro de las diferentes alternativas hay algunas que destacan por su popularidad, facilidad de uso y confiabilidad:

Windows

La familia de sistemas operativos para uso personal Windows fue desarrollada por Microsoft. Sus sistemas operativos son extremadamente populares, incluso llegando a mil millones de usuarios activos mensualmente en 2020 a través de Windows 10 [4], representando aproximadamente un 80.5 % todas las computadoras de escritorio y laptops[5] .

La última versión de Windows es Windows 11, diseñado con énfasis en facilidad de uso, seguridad y productividad[6].

macOS

El sistema operativo macOS fue el sucesor de Mac OS. Es un sistema operativo propietario usado en algunos dispositivos Apple. A pesar de su nombre actual, macOS anteriormente tenía el nombre Mac OS X y OS X[7].

Aunque menos popular que Windows, macOS es el tercer sistema operativo más popular en computadoras de escritorio y laptops; con una tasa de mercado del 7.5 %[5].

iOS

iOS es el sistema operativo diseñado por Apple para correr en dispositivos móviles. Originalmente estaba derivado Mac OS X[3]. Es un sistema operativo propietario. Entre los

dispositivos que utilizan iOS se encuentran el iPad, iPhone y Apple TV[3].

Se destaca por su interfaz de usuario amigable, su estrecha relación con el *hardware* y su integración con el resto del ecosistema de Apple.

Android

Android es un sistema operativo de código abierto para dispositivos móviles. Está basado en el sistema operativo Linux[3].

Existen diferentes versiones de Android. La versión estable más moderna al momento de escritura es Android 13. Pero, ya existe una versión beta de Android 14[8].

Una de las prioridades del sistema operativo Android es mantener la privacidad de sus usuarios. Debido a esto, tienen un modelo estricto sobre cómo otorgan a diferentes aplicaciones permisos de acceder a recursos sensibles del sistema. De la misma manera, dependiendo de la sensibilidad del recurso, Android otorgará diferentes tipos de permisos.

Permisos en el momento de instalación Dentro de esta categoría se encuentran todas las actividades que tienen poco impacto en el sistema y afectan a otras aplicaciones. Estos permisos serán otorgados a la aplicación en el momento que sea instalada[9].

Permisos de tiempo de ejecución Para que una aplicación pueda acceder a recursos potencialmente peligrosos, es necesario que se solicite un permiso en tiempo de ejecución. Este tipo de permisos se debe solicitar dinámicamente. Después de realizar la solicitud, el sistema operativo procederá a mostrarle una notificación al usuario pidiéndole su autorización; informándole sobre las posibles implicaciones de otorgar el permiso solicitado[9].

Permisos especiales Los permisos especiales son reservados para actividades críticas dentro del dispositivo. El manufacturador será encargado de decidir cuáles actividades necesitarán de permisos especiales y cuáles no[9].

5.2. *Hardware*

El *hardware* se refiere a todos los componentes físicos de una computadora[10]. El *hardware* es la estructura donde se ejecuta el *software*.

Algunos componentes importantes del *hardware* son:

5.2.1. CPU

La unidad central de procesamiento (CPU por sus siglas en inglés) es comúnmente referida como el cerebro de la computadora. En términos generales está formada por dos

partes[10]:

- **Unidad de control**: Supervisa qué operaciones van a ser ejecutadas y su orden. También está encargada de administrar la comunicación con otros componentes de la computadora.
- **Unidad lógica-aritmética**: Realiza operaciones aritméticas básicas; como la suma, resta, división y multiplicación. Asimismo, realiza diferentes operaciones lógicas como AND, OR, XOR[10].

5.2.2. Memoria RAM

La memoria RAM (*Random Access Memory*) se utiliza para almacenar las instrucciones que van a ser ejecutadas por el CPU y los datos que van a ser operados. Una característica de la memoria RAM es que es volátil; es decir, al apagarse el dispositivo se perderá toda la información que almacenada[10]. Comúnmente es utilizada como un intermediario entre el disco duro y el CPU.

5.2.3. Tarjeta madre

La tarjeta madre contiene y comunica diferentes componentes de una computadora. Dependerá de la tarjeta madre el tipo de dispositivos que se pueden conectar al computador[10].

5.2.4. Disco duro

El disco duro es un tipo de memoria, similar a la RAM. Sin embargo, los contenidos almacenados en el Disco Duro no serán perdidos al apagar la computadora[10].

A pesar de esto, la lectura y escritura al disco duro es considerablemente más lenta que la de la memoria RAM[10]. Por lo que se prefiere almacenar información temporal o comúnmente usada en la memoria RAM.

5.3. *Software*

El *software* es un conjunto de instrucciones que forman programas de computadora[11]. En términos generales, el *software* se puede interpretar como todos los aspectos lógicos de una computadora.

5.3.1. Aplicaciones móviles

Por otro lado, una aplicación es un tipo de *software* con un propósito específico[12]. Una manera de clasificar aplicaciones es el tipo del dispositivo en el que se ejecuta. Por ejemplo,

las aplicaciones móviles son aplicaciones que se ejecutan en dispositivos como teléfonos y tabletas.

5.3.2. Lenguajes de programación

El diseño y creación de *software* es un proceso complejo. A raíz de esto, se crearon diferentes lenguajes de programación para simplificar y agilizar la producción de *software*[13]. Los lenguajes de programación buscan ser un intermedio entre el lenguaje de máquina y el lenguaje natural[12].

Los lenguajes de programación pasarán por un proceso de compilación o interpretación para convertirse en instrucciones que puedan ser ejecutadas directamente por una computadora. Un buen lenguaje de programación debe facilitarle al ser humano diseñar, crear, realizar pruebas y darle soporte a diferentes piezas de *software*[13].

Usos de diferentes lenguajes de programación

Algunos lenguajes son más comúnmente utilizados para implementar ciertos tipos de aplicaciones. Esto va a depender de las características del lenguaje, su historia y la comunidad de programadores que los utiliza. No existe una clasificación formal sobre los usos de diferentes lenguajes de programación, de manera que puede considerarse en gran manera algo subjetivo.

- **Frontend (Web)**: Son utilizados para el desarrollo de páginas web y su lógica visual.
  - Javascript
- **Backend**: Comúnmente se ejecutan en servidores. Son utilizados para compartir archivos, comunicarse con bases de datos y otro tipo de lógica sensible que no puede estar ubicada en otros lugares.
  - Java
  - Python
  - Ruby
  - Javascript
- **Desarrollo móvil**:
  - Dart
  - Javascript
  - Java
  - Kotlin
  - C++
- **Automatización**:

- Python
- Ruby
- Bash
- PowerShell

- **Desarrollo de videojuegos**
  - C++
  - C#
- **Sistemas embebidos**: Los sistemas embebidos son dispositivos que comúnmente solo realizan una función, como microondas, refrigeradores, routers, etc.
  - C
  - C++
  - Rust
- **Ciencia de datos e Inteligencia Artificial**:
  - Python
  - R

5.3.3. API

Una *application programming interface* (API) es un conjunto de procedimientos que permiten la interacción de sistemas de *software* diferentes[12]. Estos sistemas pueden ser simplemente diferentes segmentos de código o incluso *software* ejecutándose dispositivos diferentes.

API Remotas

Las API remotas interactúan con aplicaciones que están siendo ejecutadas en un dispositivo diferente al que realiza la solicitud. Para que la comunicación sea exitosa, ambos dispositivos necesitan definir el protocolo que usarán para comunicarse. Uno de los protocolos más usados en las API remotas en la web es HTTP[14].

API de sistemas operativos

Los sistemas operativos generalmente proveen un API para que las aplicaciones puedan comunicarse con otros elementos del sistema. Por ejemplo, el API de Windows, Win32, le permite a una aplicación crear archivos, procesos y canales de comunicación[10].

API de código

Las API de código permite la comunicación entre dos segmentos de código diferentes[14], como un *framework*.

5.3.4. *Framework*

En *software*, un *framework* es una plantilla para el desarrollo de aplicaciones. Proveen estructuras de código que cuentan con funcionalidad básica. Los usuarios de un *framework* deberán tomar las estructuras provistas y construir sobre ella la funcionalidad que tendrá la aplicación que desean construir[12].

5.3.5. Base de datos

Una base de datos es cualquier sistema que permita guardar información. El término bases de datos es más comúnmente utilizado para hacer referencias a piezas de *software* especializadas en almacenar información, comúnmente de manera persistente[15].

Este tipo de tecnologías traen consigo las siguientes ventajas:

- Aumento de productividad
- Disminución en la redundancia de datos
- Seguridad de los datos[15]

Esto no significa que el diseño de las bases de datos actual son perfectos. En la actualidad, la mayoría de bases de datos tradicionales cuenta con las siguientes desventajas:

- Complejidad del uso
- Costos de almacenamiento
- Rigidez sobre el tipo de información que puede guardar[15]

5.3.6. Paralelismo

Gran parte del avance en el aumento de la capacidad computacional está causado por el incremento en la cantidad transistores en un CPU. Sin embargo, aumentar la cantidad de transistores en un CPU implica que se necesitará más energía para funcionar y que producirá más calor. A su vez, altas temperaturas dañan y reducen el tiempo de vida de algunos componentes electrónicos. [16].

Para solucionar esto, en vez de tratar de colocar más transistores dentro de un CPU, se empezó a diseñar CPUs formados por más de una unidad de procesamiento[16]. Este acercamiento proporcionó la habilidad de ejecutar más de una tarea al mismo tiempo.

Se refiere con paralelismo a la habilidad ejecutar más de una tarea al mismo tiempo[12]. El paralelismo es comúnmente alcanzado por un computador al ejecutar varios procesos o un proceso con varios hilos al mismo tiempo.

Procesos

Para que una aplicación se ejecute en un dispositivo es necesario que se cree un proceso. Un proceso está formado por diferentes componentes, como las instrucciones del programa, el hilo de ejecución, la memoria que se encuentra utilizando y otra información de seguridad[16].

Hilos

Los hilos de ejecución, mejor conocidos como hilos o *threads*, son partes de un proceso que contienen información como cuál instrucción se está ejecutando actualmente. Un proceso puede contar con más de un hilo de ejecución[2].

5.4. Redes de computadoras

Una red de computadoras consiste en un grupo de computadoras que son capaces de comunicarse entre sí. Esta conexión puede ser física o virtual. Uno de los ejemplos más importantes de redes de computadoras es el Internet[17].

A pesar de que existen diferentes maneras de clasificar una red de computadoras, usualmente se categorizan según su tamaño:

5.4.1. Tipos de redes

LAN - Red de área local Las redes de área local (o LAN por sus siglas en inglés), son redes pequeñas de uso privado. El espacio físico que puede abarcar este tipo de red puede ir desde una oficina pequeña hasta unos pocos kilómetros[18].

MAN - Red de área metropolitana Una red MAN es una red diseñada para abarcar decenas o incluso centenas de kilómetros. Usualmente cubren ciudades enteras. Pueden usarse para conectar varias redes LAN o como una red individual[18].

WAN - Red de área amplia Las redes WAN son redes de gran tamaño. Su alcance físico es de países o incluso el mundo entero. El internet está clasificada como una red WAN de alcance global[18].

5.4.2. Direcciones IP

Las direcciones IP son parte del Protocolo de Internet (IP por sus siglas en inglés). Las direcciones IP se usan para representar a cada dispositivo conectado a una red[18].

Existen dos tipos de direcciones IP utilizadas actualmente, las direcciones IPv4 y las direcciones IPv6:

IPv4

Las direcciones IPv4 están formadas por cuatro octetos[18]. Comúnmente se representa a cada octeto en sistema decimal y se separa a los diferentes octetos por un punto. De manera que, en notación decimal, todas las direcciones IPv4 están en el rango de 0.0.0.0 hasta 255.255.255.255. Esto representa un total de 2^{32} (o 4,294,967,296) direcciones IPv4 únicas. Al ser el formato de direcciones IP más usado en la actualidad, es la que se tiende a hacer referencia cuando únicamente se brinda el término dirección IP.

IPv6

Por otro lado, las direcciones IPv6 cumplen los mismos objetivos que las direcciones IPv4, pero solucionan diferentes problemas existentes en el protocolo IPv4; cuentan con un encabezado más simple y ofrecen un mejor soporte para tránsito de archivos multimedia. IPv6 también provee considerablemente más direcciones IP que IPv4. Esto es debido a que IPv6 utiliza 128 bits, a diferencia de los 32 usados en IPv4[19][20]. En total, existen 2^{128} (exactamente 340,282,366,920,938,463,463,374,607,431,768,211,456) direcciones IPv6 únicas.

Sin embargo, a pesar de todas las ventajas ofrecidas por IPv6, su adopción ha sido relativamente lenta y la mayoría del tráfico red sigue usando IPv4[21].

5.4.3. Subredes

Por razones de facilidad, eficiencia y seguridad, es conveniente aislar redes innecesariamente grandes en redes pequeñas[22][18]. A estas redes dentro de otras redes se les llaman subredes. Asimismo, es posible tener subredes dentro de otras subredes.

Las subredes proveen diferentes ventajas:

- Aumentan la eficiencia. Un paquete con un origen y destino en la misma subred no necesita comunicarse con la red entera[22].
- Evita problemas con asignaciones de direcciones IP que pueden ocurrir cuando una red crece de tamaño[18].
- Aislan los dispositivos entre diferentes subredes, brindando una capa adicional de seguridad[23].

Asignación de direcciones IP

Las direcciones IP no solo ayudan a identificar cada dispositivo, si no también a determinar qué ruta debe tomar cada paquete para llegar a su destino. Para lograr esto, cada red o subred es asignada un grupo de direcciones IP que son repartidas a los dispositivos conectados a esa red. De este grupo de direcciones IP, la subred reserva dos direcciones IP para usos especiales.

La primera IP asignada a una subred se usa para representar a la subred; mientras que la última dirección IP asignada se utiliza como una dirección de difusión. La dirección de difusión será usada cuando se requiera enviar un paquete a todos los integrantes de la misma red[24].

Notación CIDR

El enrutamiento entre dominio sin clases (o CIDR por sus siglas en inglés) es una manera de representar las direcciones IP asignadas a cada red. Esta notación muestra la dirección IP asignada a la red y la cantidad de bits que todos los miembros de esa subred deben tener en común con la red a la que pertenecen[24]. Vale la pena mencionar que estos bits son medidos de izquierda a derecha.

Por ejemplo, al asignarle a una red el grupo de direcciones IP de 192.168.1.0 (11000000.10101000.00000001.00000000) hasta 19.168.1.255 (11000000.10101000.00000001.11111111), se puede notar que los primeros 24 bits se mantienen iguales. En notación CIDR, la red puede ser representada como 192.168.1.0/24.

Máscaras

Otra manera de representar la cantidad de bits que definen a la red es a través de máscaras[23]. La máscara es representada como cuatro octetos, donde los primeros bits de izquierda a derecha son 1[23].

Para ilustrar, si se cuenta con la red 192.168.1.0/24, entonces los primeros 24 bits de la máscara serán 1. De manera que la máscara de esta red sería 11111111.11111111.11111111.00000000, o su equivalente en notación decimal 255.255.255.0.

NAT

Se mencionó con anterioridad que generalmente se usa IPv4 y que esta tiene un máximo de aproximadamente 4 mil millones de direcciones IP. Sin embargo, esto puede parecer un poco contradictorio. Ya se había mencionado que existen al menos mil millones de usuarios activos que usaban Windows 10[4]. Si se toma en cuenta la cantidad de teléfonos, consolas de videojuegos y otros dispositivos inteligentes, claramente no existen suficientes direcciones IPv4 para que cada dispositivo tenga una dirección única.

Aunque la solución para la falta de cantidad de direcciones IPv4 es migrar completamente

a IPv6, este es un proceso que tomará varios años. De manera que, como solución a corto plazo, se creó un nuevo protocolo llamado *NAT*.

La traducción de direcciones de red (NAT por sus siglas en ingleś) está descrita en el documento RFC3022[20]. NAT funciona de manera análoga a como se trabaja con direcciones físicas en edificios con varias oficinas o apartamentos. Al rededor del mundo, existe una infinidad de apartamentos con el mismo número, pero todos se distinguen por la dirección del edificio.

Igualmente, en el caso de NAT, cada subred va a tener su dirección IP única y todos los dispositivos conectados serán asignados su propia dirección IP interna. De esta manera, aún si dos dispositivos comparten la misma dirección IP interna, se pueden diferenciar por la dirección IP de la red o subred a la que pertenecen[20].

IPs Privadas

Para facilitar el uso de NAT, un conjunto de direcciones IPs fueron reservados únicamente para uso privado[20]. Volviendo a la comparación con un hotel, se apartaron estas direcciones IP para ser el análogo al número de apartamento.

Los rangos de direcciones IP privadas son:

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255[25]

5.4.4. Dominios

La comunicación en internet está basada en direcciones IP. Sin embargo, las direcciones IP son complejas de memorizar para el ser humano. Para volver al uso de internet más amigable, se diseñó un sistema donde direcciones IP y nombres de dominio pueden intercambiablemente en la mayoría de los casos.

Los nombres de dominio son únicamente cadenas de texto alfanuméricos[26] que pueden ser traducidas a direcciones IP.

Por ejemplo, si se desea navegar a la página de uvg.edu.gt, no se necesita ingresar su dirección IP (por ejemplo 13.68.205.253), basta con ingresar el nombre uvg.edu.gt y este será traducido a su dirección IP respectiva.

5.4.5. Protocolos

Los protocolos son simplemente un conjunto de reglas que indican cómo se debe llevar a cabo la comunicación; especifican detalles como qué información se va a compartir, cómo se va a compartir, en qué formato se va a compartir y cuándo se va a compartir[20].

El concepto de protocolos no se encuentra limitado a aspectos técnicos, también se puede usar en contextos humanos.

RFC

Originalmente, los primeros documentos RFC fueron llamados solicitud de comentarios, pero ahora son principalmente conocidos por sus siglas en inglés, RFC[27].

Los RFC describen varios aspectos importantes para las redes de computadoras. Pueden especificar estándares importantes usados en redes como TCP, UDP, NAT, DNS, entre otros[27].

5.4.6. Capas OSI

Una de las maneras para definir el proceso realizado para la comunicación en computadoras es el modelo *Open System Interconnection* (OSI). Este modelo define una serie de siete capas aisladas con diferentes responsabilidades[24].

Cada capa cuenta con sus propios protocolos y funcionalidades específicas. Asimismo, todas deben seguir los siguientes principios:

1. Cada capa debe tener una función bien definida[18].
2. La función de cada capa debe considerar protocolos existentes[18].
3. Las capas deben minimizar la cantidad de información transmitidas con las otras capas[18].

Capa física

La capa física consiste en el intercambio de información en un medio físico. Únicamente está encargada del intercambio de bits[18]. Esta actúa como un puente entre los diferentes factores físico-mecánicos del transporte de datos y otros aspectos relacionados al *software*.

La capa física traduce diferentes señales físicas a un conjunto de bits y viceversa. Dichas señales pueden provenir de medios como cables y microondas[20].

Esta capa es susceptible a varios tipos de interferencias. Por ejemplo, puede que otros dispositivos afecten la transmisión de otros paquetes que utilicen las mismas frecuencias.

Capa de enlace de datos

La capa de enlace está encargada de transmitir información entre dos computadoras conectadas directamente. Esto implica que tiene la responsabilidad de determinar el inicio y final de cada paquete recibido en la capa física. También debe controlar la velocidad a

la que se transmiten la información; puede que el receptor no sea capaz de interpretar la información recibida con la misma velocidad del emisor[20].

La capa de enlace de datos tiene una segunda función importante: verificar que la información transmitida en la capa física[20]. Esto es realizado a través de diferentes algoritmos de detección y corrección de errores como checksums, bits de paridad y CRC[23].

Datagramas IP Los datagramas de IP definen cómo se va a pasar información en la capa de enlace. Están compuestos de dos partes: el encabezado (contiene metadatos de la información transmitida) y el área de texto (contiene el contenido que se será comunicado al receptor)[18].

Entre los metadatos en el encabezado se encuentra información como la cantidad máxima de tiempo para que el paquete termine su trayecto, el tamaño de la información transmitida, la dirección IP del emisor y la dirección IP del receptor[18].

Capa de red

La capa de red tiene la única responsabilidad de buscar a dónde se tendría que enviar la información para que llegue a su destino[23]. Esta capa es principalmente implementada en *routers*.

El envío de paquetes cuando el emisor y receptor están en la misma red es relativamente simple. En la mayoría de casos, se puede enviar el paquete de manera directa al receptor[20].

Por otro lado, si el destinatario de un paquete se encuentra en otra red, será necesario que el paquete viaje a través de uno más intermediarios hasta llegar a su destino. Esto únicamente es posible si todos los intermediarios respetan y calculan correctamente a dónde deberían redirigir cada paquete[20].

La manera en que cada parada decide cuál interfaz usar para reenviar el paquete es a través de diferentes algoritmos de ruteo[23].

Existen diferentes maneras de catalogar los diferentes algoritmos de ruteo:

- Algoritmos centralizados y no centralizados:
  - Los algoritmos centralizados necesitan tener conocimiento completo de la red para determinar cuál es el camino más óptimo para cada paquete. Cabe mencionar que dependiendo del tamaño de la red, puede que el uso de estos algoritmos no sea factible[23].
  - En los algoritmos no centralizados cada nodo solo tiene conocimiento de su vecino inmediato. De esta manera, para determinar cuál es la ruta óptima, cada parada debe realizar cálculos para obtener un aproximado de cuál vecino directo es parte de la ruta óptima al receptor final[23].
- Algoritmos dinámicos y estáticos

- Los algoritmos dinámicos actualizan constantemente la ruta óptima para llegar a cada destino. Toman en cuenta aspectos como problemas en otros nodos y congestiones de red[23].

- En el caso de los algoritmos estáticos, raramente se van a considerar cambios en la red. Por el contrario, comúnmente los únicos cambios provocados a las rutas óptimas tienden a ser ocasionados manualmente por un ser humano[23].

Capa de transporte

La capa de transporte oculta toda la complejidad que existe en las capas anteriores; como el proceso de ruteo y corrección de errores[18]. Esta capa representa un canal de comunicación lógico[23].

Esta capa introduce un concepto importante para la comunicación entre redes, los puertos de comunicación. Los puertos son un concepto abstracto que permiten identificar qué proceso está enviando y recibiendo un mensaje[20]. Algunos puertos están reservados para protocolos importantes. Por ejemplo, el puerto 587 suele utilizarse para *Simple Mail Transfer Protocol* (SMTP) o correo.

TCP El protocolo TCP (*Transmission Control Protocol*) trata de asegurarse que toda la información sea transmitida exitosamente en el orden que fue enviada[24]. Para lograr esto, el receptor debe confirmar que cada paquete fue entregado con éxito. De manera que, ambos lados de la comunicación deben llevar la cuenta de cuántos paquetes se han intercambiado y el tamaño de cada uno. Este es un aspecto importante, ya que implica que tener una conexión TCP abierta usará recursos, aún si no se envían paquetes[23].

Otra de las maneras que utiliza TCP para tratar de mejorar las conexiones es asegurarse que el emisor y receptor están listos para intercambiar información. Esto lo hace empezando cualquier canal de comunicación entre dos sistemas con un *Three-Way Handshake*[28].

Three-Way Handshake El *Three-Way Handshake* consiste en una comunicación de tres mensajes, donde el emisor y receptor negocian algunos aspectos de la conexión y demuestran que están listos para comunicarse.

UDP Por otro lado el protocolo UDP (*User Datagram Protocol*) no garantiza que un paquete llegue a su destino. Esto lo vuelve un canal inseguro, pero más simple[23]; proporcionando sus propias ventajas y desventajas.

Entre sus diferencias se encuentra de que no confirma de que el receptor esté listo para la recepción de mensajes. En otras palabras, no necesita realizar el *Three-Way-Handshake*[28].

La simpleza del protocolo UDP lo vuele más rápido que el protocolo TCP. Esto lo vuelve más usado en situaciones donde se prefiere que los paquetes lleguen rápido a la seguridad de que los paquetes llegaron; por ejemplo, en transmisiones de video[28].

Capa de sesión

La capa de sesión se encarga de sincronizar mensajes y conectar al emisor y receptor lógicamente para permitir el paso de mensajes[12].

Capa de presentación

La capa de presentación se encuentra muy relacionada a la capa de aplicación. En esta capa, se codifica la información en tránsito a un formato que la aplicación sea capaz de interpretar[23].

Esta capa también proporciona otros servicios como encripción y compresión de datos[23].

Capa de aplicación

En la última capa OSI se encuentran todas las aplicaciones y protocolos relacionados al tráfico de red[23].

Algunos de los protocolos y aplicaciones más importantes para el uso del internet moderno son:

DNS *Domain Name Service* o DNS es una de las aplicaciones más importantes de la actualidad. Está encargada de traducir dominios a direcciones IP concretas[23]. Esto lo realiza a través una jerarquía de servidores. De manera que si un servidor no contiene la información que se está buscando, la busca en un servidor DNS superior[23].

El servicio DNS se comunica a través de UDP en el puerto 53[23].

HTTP y HTTPS El protocolo *Hypertext Transfer Protocol* (HTTP) es uno de los protocolos dominantes en el internet actual[23]. Es un formato simple y extendible que facilita el transporte de diferentes archivos, como archivos HTML[29]. Similarmente, el protocolo *Hypertext Transfer Protocol Secure* (HTTPS) es el protocolo HTTP pero encriptado a través de SSL[23].

HTTP URL Un *Uniform Resource Locator* (URL) es un estándar para localizar recursos en diferentes aplicaciones. Tiene soporte para diferentes protocolos como FTP, Gopher y HTTP[30].

Una URL HTTP está formada de las siguientes partes:

- Protocolo: En este caso, HTTP
- Servidor: Comúnmente representado como un dominio.

- Puerto: Especifica el puerto al que se va a conectar el cliente. De no ser especificado, se toma el puerto default de 80 (HTTP) o 443 (HTTPS).
- Recurso: Muestra la ubicación del recurso al que se desea acceder.
- *Query*: Otro tipo de información que ayude al destino a identificar qué recurso se está buscando.

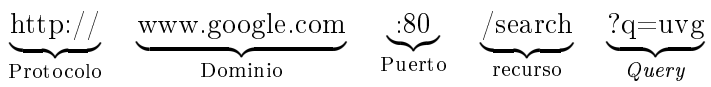


Figura 1: Estructura de un HTTP URL

Whois El protocolo Whois es utilizado para proveer información sobre quién es el dueño de un dominio. Cabe mencionar Whois se ha expandido para cumplir con necesidades legales y de propiedad del país en el que se encuentra el servidor.[31].

5.4.7. *Firewall*

Un cortafuegos o *Firewall* es un dispositivo o aplicación encargada de bloquear o permitir el paso de tráfico red[23].

Un *Firewall* tiene tres objetivos:

- Supervisar todos los paquetes entrantes y salientes de la red
- Únicamente permitir el tráfico que se adhiera a las políticas configuradas
- Ser impenetrable[23]

5.4.8. VPN

Una red privada puede ser usada para transmitir información de manera más privada y segura. Sin embargo, las redes privadas raramente son factibles cuando se intenta conectar dispositivos en diferentes localizaciones geográficas. Para solucionar este problema se crearon las redes privadas virtuales (VPN por sus siglas en inglés)[23].

Al usar una VPN, el tráfico red de un dispositivo será encriptado y enviado a la red destino. En la red destino, los paquetes serán desencriptados y tratados como si se hubieran originado directamente de esa red[32].

Asimismo, una VPN puede afectar modificar algunos ajustes sobre la configuración del dispositivo. Entre estos detalles se encuentran:

- Dirección IP a la que responderá el dispositivo

- DNS que utilizará el dispositivo
- Cantidad máxima de tiempo que se debe esperar por un paquete
- Etcétera

5.5. Ciberseguridad

La ciberseguridad es un conjunto de técnicas para proteger información electrónica. Su objetivo principal es garantizar la confidencialidad, integridad y disponibilidad de la información digital[33].

Entre estas técnicas utilizadas por la ciberseguridad se encuentra la administración de riesgo, la administración de incidentes y la administración de identidad. Tener buenos sistemas de ciberseguridad implica tener un buen conocimiento sobre amenazas potenciales y las áreas vulnerables de una organización[33].

5.5.1. Amenazas

Las amenazas a la ciberseguridad son todos los eventos que podrían afectar negativamente a un sistema[33]. Las amenazas no necesariamente son ocasionadas con malas intenciones; pueden haber sido creadas por error humano.

Malware

El *malware* o aplicación maliciosa es un tipo de *software* que busca dañar, afectar el uso de otras aplicaciones o incluso robar información confidencial[34]. El *malware* comúnmente es creado con objetivos financieros o de odio.

Phishing El *phishing* es un tipo de *malware* que busca robar información sensitiva; comúnmente credenciales. Este tipo de ataques busca hacerse pasar por otra entidad, engañando a la víctima para que proporcione información privada[34].

Este tipo de *malware* se destaca porque en vez de buscar vulnerabilidades en el *software*, trata de engañar al humano.

Ransomware El *ransomware* es un tipo de virus que destruye o bloquea el acceso a información crítica. Luego, le pide a la víctima que pague un rescate con la promesa de que los datos serán desbloqueados[34].

5.5.2. Dominios maliciosos

No todas las páginas web son seguras. Para tratar con esto, existen muchos sistemas para detectar y bloquear el acceso a páginas maliciosas. Esto ha provocado que los cibercriminales registren miles de dominios para la misma página web. De esa manera, aún si la mayoría es detectada como maliciosa, simplemente se cambian a un dominio que no haya sido detectado como malicioso[35]. Como resultado de esto, aproximadamente un 20 % de todos los dominios registrados en el primer semestre de 2022 fueron registrados como maliciosos[35].

5.5.3. El factor humano en la ciberseguridad

En el área de la ciberseguridad generalmente se reconoce que el humano es el eslabón más débil. Aún con el mejor sistema de ciberseguridad, si el usuario sigue las peores prácticas, el sistema puede llegar a ser trivial de penetrar. Por ejemplo, en uno de los ataques cibernéticos más famosos, el ataque de Anonymous a Hbgary, fue causado por vulnerabilidad humana. La falta de conocimientos y malas prácticas humanas de ciberseguidad fueron los principales puntos de entrada para el ataque[36].

Educación sobre ciberseguridad

Debido a la escacez de profesionales capacitados sobre ciberseguridad, existen muchos recursos que buscan educar sobre el tema[37]. Por ejemplo, el proyecto *Be Internet Awesome* para educar niños sobre ciberseguridad[38].

Educar sobre ciberseguridad no solo involucra modificar los hábitos de las personas. También involucra modificar la mentalidad y valores para buscar una protección proactiva.

5.5.4. El estado actual de la ciberseguridad en Android

En el 2019 se registraron 2.5 billones de dispositivos Android activos[39]. A pesar de que esta cifra incluye a dispositivos como televisores, relojes e incluso vehículos, todavía sirve para reflejar que existe una abundancia de teléfonos Android circulando.

Al ser una plataforma muy popular, una cantidad significativa de *malware* son desarrollados para Android. En febrero de 2022, se determinó que la cantidad de nuevos *Malwares* desarrollados para Android aumentó un 500 %[1]

Otro aspecto que vuelve especialmente vulnerable a los teléfonos Android es su habilidad de descargar aplicaciones provenientes fuera de la Play Store. Aunque esto provee más libertad a sus usuarios, también los pone en riesgo. La Play Store provee una capa adicional de protección al evitar que ciertas aplicaciones maliciosas estén disponibles[40].

5.6. Inteligencia Artificial

Existen dos tipos de definiciones sobre la inteligencia artificial. En la primera definición, se define a la inteligencia artificial como un agente que actúa con comportamiento humano. Por otro lado, también se puede medir la inteligencia artificial según la eficiencia en la que realiza cierta acción[41].

5.6.1. Modelos de inteligencia Artificial

Por otro lado, los modelos de inteligencia artificial son programas reconocen patrones e toman una decisión[42]. Un modelo puede estar implementado en código, circuitos, fórmulas matemáticas, entre otros[41].

Redes neuronales

Los primeros modelos de inteligencia artificial trataban de imitar al cerebro. El cerebro está compuesto de neuronas interconectadas. Las neuronas son células que reciben, transforman y propagan señales electro-químicas[41].

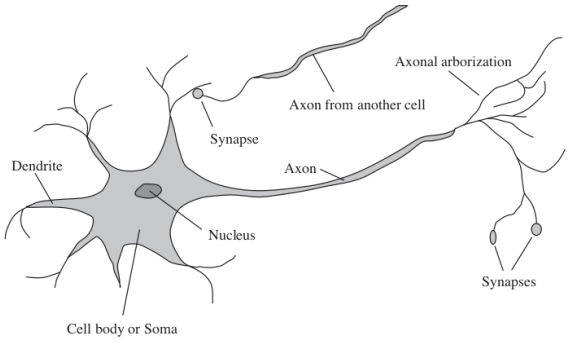


Figura 2: Estructura de una neurona[41]

Sin embargo, una neurona individual no es suficiente para tomar una decisión. El verdadero valor de una neurona solo se muestra cuando se conectan varias neuronas unas a las otras. Se le conoce como redes neuronales al conjunto de neuronas interconectadas[41].

Los modelos de inteligencia artificial de redes neuronales imitan estas características a través de *software*.

Árboles de decisión

Los árboles de decisión son uno de los modelos de inteligencia artificial más simples. Consiste en un grupo de decisiones que determinan la acción resultante.

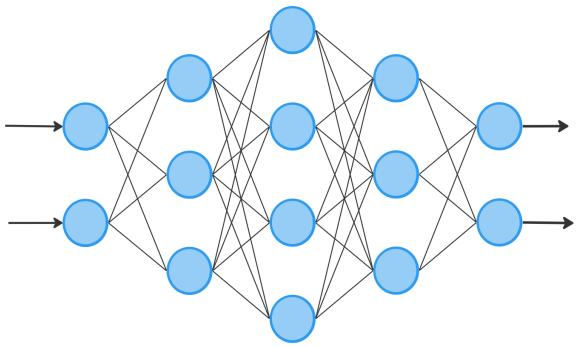


Figura 3: Red neuronal

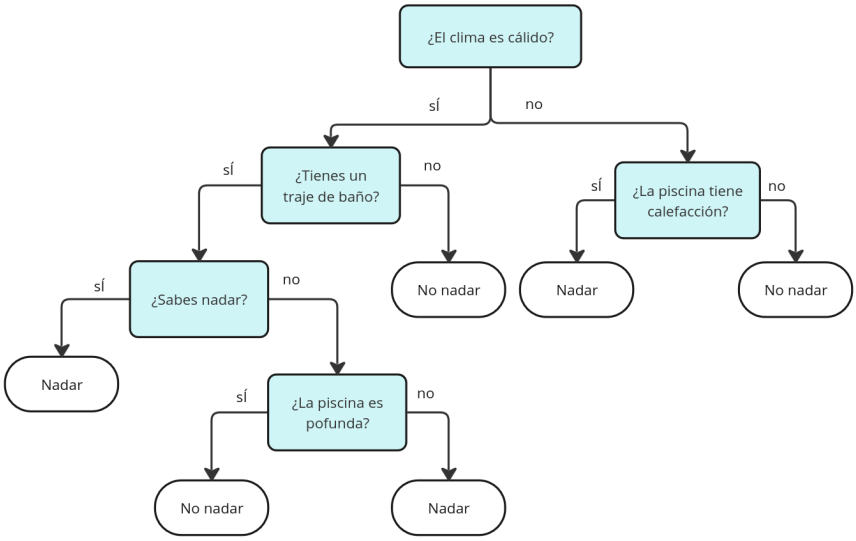


Figura 4: Árbol de decisiones sobre nadar o no nadar

SVM

Los modelos SVM o *Support Vector Machine* funcionan a través de planos multidimensionales. Luego, codifican los datos de entrada del modelo para que puedan ser colocados en este espacio multidimensional[41].

Finalmente, los modelos SVM dividen ese espacio multidimensional en dos. La decisión del modelo estará basada sobre de qué lado de la división se encuentran los nuevos datos obtenidos[41].

5.7. Metodologías de desarrollo

5.7.1. Scrum

Scrum es una metodología de desarrollo donde se busca entregar avances iterativos. Aunque comúnmente es utilizada para el desarrollo de *software*, es posible utilizarla en otros

ambientes de trabajo[43].

Sprints

Los *sprints* son periodos de tiempos cortos (usualmente 2-3 semanas) donde el equipo se compromete a cumplir unas tareas. Los *sprints* pueden estar centrados en cumplir una meta, lo que afecta a la selección de tareas que se seleccionan para cada *sprint*[43].

Roles

- ***Product Owner***: Es la persona encargada de pensar en lo que le beneficia al cliente. Para esto, deben determinar cuáles tareas son las más importantes para orientar al equipo.
- ***Scrum Master***: Supervisan el proceso de Scrum. buscan mejoras en los procesos de desarrollo.
- **Equipo de desarrollo**: Desarrollan y entregan el trabajo.

Ciclo de trabajo Scrum

Antes de iniciar cualquier *sprint*, se debe definir con el apoyo del *Product Owner* y el equipo de desarrollo cuales tareas se realizarán. Esto se hace durante reuniones llamadas *Sprint Planning*[43].

Una vez se definieron las tareas del *sprint*, el equipo de desarrollo trabajará a lo largo del *sprint* para cumplir con las tareas asignadas. Durante este lapso de tiempo, se tendrá una reunión diaria con el *Scrum Master* para revisar el avance de los miembros e identificar cualquier bloqueo[43].

Cuando termine el *sprint*, se tendrá una reunión informal llamada *sprint review*, donde se mostrará una demo de los avances realizados a las partes interesadas. De manera que el equipo pueda obtener retroalimentación temprana[43].

Finalmente, se finaliza el *sprint* en una reunión retrospectiva, donde se documenta qué aspectos funcionaron del proyecto y cuáles requieren de mejoras[43].

5.7.2. Kanban

La metodología de trabajo Kanban está enfocada en optimizar el flujo de trabajo. Está centrado en visualizar el trabajo realizado a través de un tablero, donde todos los miembros del equipo documentan el estado de las tareas en las que están trabajando.

Una de las ventajas más grandes de Kanban es su flexibilidad. Al no tener iteraciones definidas, el equipo puede trabajar continuamente en las tareas pendientes. Asimismo, es adaptable a cambios; añadir nuevas tareas no afectará al resto de miembros.

Tablero Kanban

El tablero Kanban es una representación gráfica donde todos los miembros del equipo pueden ver en tiempo real sobre el avance de cada miembro del equipo[44].

El tablero está dividido en columnas que representa el estado de las diferentes tareas. Puede existir cualquier cantidad de columnas según el equipo considere necesario, pero puede ser tan simple como Pendiente, En progreso y Completado[44].

CAPÍTULO 6

Metodología

Se realizarán dos tipos de pruebas con los usuarios: pruebas de la usabilidad de corto plazo, pruebas de usabilidad en periodos más largos de tiempos y pruebas de aprendizaje.

6.1. Pruebas de usabilidad

6.1.1. Pruebas de corto plazo

Las pruebas de usabilidad iniciales serán de corta duración. Generalmente, no más de 20 minutos. Únicamente se le mostrará el último prototipo que se ha desarrollado. El usuario usará el prototipo por unos minutos, se observarán sus reacciones y se pedirá su opinión. Tendrá naturaleza informal; su único objetivo es validar qué opina el usuario de la apariencia de la aplicación.

6.1.2. Pruebas de largo plazo

El segundo tipo de pruebas de usabilidad tendrá una duración de tiempo más larga. Serán lanzadas cuando las funcionalidades de la aplicación ya estén completas o casi completas.

El usuario tendrá acceso completo a la aplicación por al menos dos semana. Se acompañará al usuario recopilando sus impresiones de la aplicación a lo largo del tiempo de la prueba. Con este tipo de pruebas se busca evaluar la experiencia de usuario de la aplicación entera y tener acceso a un nivel de retroalimentación que no sería posible en una prueba rápida.

6.2. Pruebas de aprendizaje

Las pruebas de aprendizaje tienen dos objetivos: verificar cuánto ha aprendido el usuario sobre ciberseguridad y determinar si mostrar una explicación sobre el tipo de dominio bloqueado sí mejora la probabilidad de que un usuario se defienda exitosamente ante un ataque.

Para lograr esto, se trabajará con dos grupos de participantes:

- Participantes de control: Aunque tendrán acceso a Ela, cuando se bloquee la comunicación a un dominio malicioso, no recibirán información sobre el tipo de virus detectado. Solo serán informados de que se bloqueó el acceso a un dominio.
- Participantes de prueba: Este segundo grupo de participantes tendrá una versión de Ela diferente a la usada por el primer grupo. Cada vez que se bloquee el acceso a un dominio malicioso; la aplicación les mostrará una descripción sobre qué tipo de ataque fue y que acciones podrían tomar para protegerse.

Ambos grupos recibirán un cuestionario que buscará medir sus conocimientos de ciberseguridad. Este cuestionario será enviado al inicio y al final de la prueba.

Posteriormente, se procederá a realizar un simulacro de un ataque de ciberseguridad, donde se comparará entre ambos grupos cuál de ellos manejó de mejor manera los ataques simulados.

6.3. Metodología de desarrollo

Como metodología de desarrollo se optó por usar una mezcla entre Kanban y Scrum. Al inicio de cada *sprint* se determinaría una meta y se asignarían las tareas relevantes para el trabajo. La razón por la que se decidió usar una mezcla entre Kanban y Scrum, fue porque se buscaba la flexibilidad de Kanban, pero se consideró que el concepto de *sprints* podría agregar valor para planear un cronograma de desarrollo.

Al inicio de cada *sprint* se determinaría un objetivo y se asignarían las tareas relevantes. Cada *sprint* tendrá una duración máxima de 3 semanas, pero puede durar menos tiempo si se completan todas las tareas asignadas. Si no se habían terminado todas las tareas del *sprint* después de llegar al límite de tiempo, serían reasignadas en el siguiente *sprint*.

Por otro lado, se decidió que no se deberá tener más de dos tareas en ejecución al mismo tiempo. De lo contrario, se perdería tiempo en lo que se cambiaba entre tareas.

CAPÍTULO 7

Resultados

7.1. Elección de herramientas

La selección de las herramientas adecuadas es un aspecto vital para el diseño de cualquier *software*. A raíz de esto, se busca elegir herramientas que puedan satisfacer las necesidades del proyecto y provean la flexibilidad de adaptarse a cualquier cambio de requerimientos que pueda ocurrir en el futuro.

7.1.1. Elección de *framework*

El *framework* es uno de los aspectos más importantes que se deben de tener en cuenta para el diseño de *software*. Este influirá fuertemente en algunas características de la aplicación final como los recursos utilizados, tiempo de desarrollo, seguridad y escalabilidad.

A raíz de todo lo anterior, se decidió enfocarse en las siguientes características para la selección del *framework*:

- **Uso de batería**: Al ser una aplicación que se mantendrá en ejecución constante, se busca limitar los recursos utilizados. De lo contrario, un usuario podría considerar que la aplicación utiliza muchos recursos y decidir desinstalarla.
- **Interacción con elementos relevantes del sistema Android**: El *framework* deberá proveer los medios para realizar las funcionalidades de la aplicación. La supervisión de tráfico de red es una funcionalidad muy específica. De manera que no está garantizado de que el *framework* tendrá proverá soporte para realizar estas actividades.
- **Estabilidad de la herramienta**: La mayoría de *frameworks* son tecnologías gratuitas

desarrolladas por personas interesadas. Aunque poco común, sí han ocurrido situaciones donde un *framework* pierde interés público o los desarrolladores deciden dejar de trabajar en ella. Para facilitar el mantenimiento de la aplicación a mediano y largo plazo, es conveniente que el *framework* se mantenga activo y reciba actualizaciones importantes, como actualizaciones de seguridad.

Para la selección de *frameworks* se tomaron a los tres *frameworks* más populares para el desarrollo de aplicaciones en Android:

- **Android SDK**: Android SDK es la manera oficial para desarrollar aplicaciones en Android. Esto significa que todos los demás *frameworks* dependen directamente o indirectamente de él. Para utilizar el SDK de Android se necesita utilizar a Java o Kotlin como lenguaje de programación. Sin embargo, es posible utilizar otros lenguajes de programación como C utilizando una extensión llamada Android NDK.

- **Flutter**: Flutter es un *framework* que utiliza el lenguaje de programación Dart. Flutter tiene una gran comunidad, lo que implica que tiene una buena cantidad de librerías y paquetes que podrían ser útiles para la aplicación. Adicionalmente, Google se encuentra apoyando el crecimiento de Flutter. Esto proporciona seguridad de que Flutter continuará recibiendo actualizaciones aún si la comunidad de desarrolladores pierde interés en Flutter.

- **React Native**: React Native es una herramienta que provee una capa de abstracción entre los componentes nativos del sistema. Utiliza el lenguaje de programación javascript, el mismo lenguaje de programación utilizado en páginas web. Esto le permite beneficiarse directamente de algunas partes del ecosistema de librerías web. Una de sus ventajas principales es que código escrito en React Native puede ser ejecutado en distintos ambientes como Android, iOS y Web con cambios mínimos.

A pesar de las ventajas y desventajas proveídas por estos *frameworks*, se optó por usar Android SDK. Al no tratar de ser compatible con otras plataformas como Web ni iOS, realiza varias optimizaciones que no son posibles en las otras dos plataformas. Esto le permite utilizar menos recursos y consumir menos energía que las otras dos alternativas.

Framework	CPU %	Máxima memoria utilizada	Uso de batería
Android SDK	2.6	72 Mb	56.6 mAh
Flutter	5.6	106 Mb	69.2 mAh
React Native	12.1	128 Mb	78.7 mAh

Cuadro 1: Recursos utilizados para renderizar una lista dinámica de 100 objetos[45]

Adicionalmente, Android SDK está respaldado directamente por Android. Para que Android pueda tener éxito, es necesario que le brinden a terceros las herramientas puedan crear nuevas aplicaciones que dependan de su ecosistema. Esto significa que el funcionamiento correcto de Android SDK es una de las prioridades de los desarrolladores de Android, garantizando que la plataforma es confiable y que no desaparecerá en el futuro cercano.

Finalmente, Flutter y React Native no cuentan con los paquetes necesarios para el tipo de aplicación que se desea implementar. Aunque cuentan con paquetes como *react-native-ip-sec-vpn*, *react-native-simple-openvpn* y *flutter_vpn* la mayoría parece estar abandonada o cuenta con otras limitaciones importantes

7.1.2. Elección del lenguaje de programación

Aunque la mayoría de *frameworks* únicamente permite utilizar un lenguaje de programación, Android SDK permite utilizar varios lenguajes como Java, Kotlin y C.

- **C**: Es uno de los lenguajes de programación más rápido que existen. Pero es considerablemente complejo de utilizar correctamente. El manejo de memoria debe realizarse de manera manual, lo que implica que a comparación con otros lenguajes, se debe tener más cuidado para evitar fugas de memoria y vulnerabilidades de seguridad derivadas de la memoria.
- **Java**: Aunque no tan rápido como C, sigue siendo uno de los lenguajes de programación más rápidos. A diferencia de C, la memoria no se maneja directamente; facilitando el diseño de programas escritos en el lenguaje. Java es ejecutado en la *Java Virtual Machine* (JVM), lo que le permite tener alto rendimiento, seguridad y retrocompatibilidad.
- **Kotlin**: Es el lenguaje recomendado para usar Android SDK. Asimismo, también es ejecutado en la JVM, brindándole aproximadamente la misma velocidad de Java. También es completamente compatible con Java, por lo que es posible utilizarlos intercambiablemente dentro del mismo proyecto. Kotlin es considerablemente más amigable que Java y C.

Se decidió utilizar el lenguaje de programación Kotlin. Aunque Kotlin no es tan rápido como C, es mucho más amigable por lo que el proceso de desarrollo será más simple. Adicionalmente, al ser el lenguaje recomendado para el desarrollo en Android SDK, la mayoría de la documentación de Android SDK está orientada a Kotlin; facilitando aún más el proceso de desarrollo.

7.1.3. Elección de base de datos

Debido a la naturaleza de la aplicación, la base de datos únicamente tiene un rol secundario. De manera que se tienen requisitos relativamente simples para la selección de la base de datos. Estos requisitos incluyen:

- Poder ser ejecutadas y almacenadas localmente en el dispositivo móvil
- Soporte para múltiples escritura y lecturas al mismo tiempo.
- Facilidad de uso

Las herramientas más populares para almacenamiento de datos persistentes en Android son:

- **SQLite**: SQLite es una base de datos de tamaño pequeño. Puede manejar la escritura y lectura en paralelo, pero es relativamente rígida sobre el tipo de información que puede almacenar.
- **Room**: Room técnicamente no es una base de datos, es una librería que utiliza SQLite. A pesar de que usar Room es aún menos flexible que usar SQLite de manera directa, es más fácil de usar. Room también incluye varias verificaciones de seguridad para proteger la integridad de los datos almacenados.
- **Datastore**: Es una librería especializada en almacenar poca información. Permite lectura y escritura al mismo tiempo (paralelismo). Cabe mencionar que internamente la información será almacenada en el dispositivo como un archivo Json o XML.

Considerando las ventajas y desventajas de las alternativas mostradas, se decidió utilizar Room y Datastore. Datastore sería utilizada para almacenar información de configuración debido a su eficiencia almacenando pequeños grupos de información.

Por otro lado, Room será utilizado para almacenar cualquier otro tipo de información, como datos sobre qué páginas fueron bloqueadas. Se preferió usar Room a SQLite debido a que no se necesitaba de una base de datos muy compleja. De manera que la libertad proporcionada por usar SQLite sin Room no proporcionaba valor a la aplicación, únicamente aumentaba la complejidad del proyecto.

7.2. Proceso de desarrollo

7.2.1. Cronograma

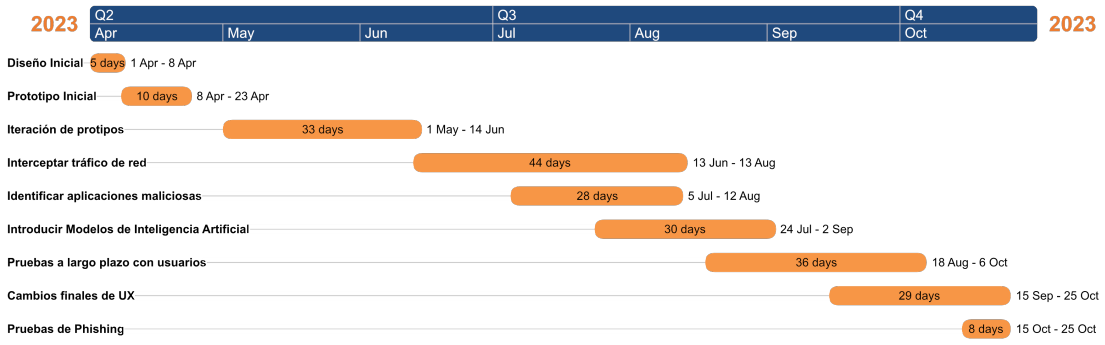


Figura 5: Cronograma

7.2.2. Sprints

Sprint 1 El objetivo del primer sprint fue la selección de aspectos gráficos la aplicación.

- Selección de colores utilizados en la aplicación
- Establecer cuál tipografía será utilizada en los diferentes títulos
- Establecer qué pantallas serán utilizadas

- Definición del flujo de la aplicación
- Prototipos en papel de la aplicación

Sprint 2 El objetivo del segundo sprint fue determinar cuáles tecnologías serán necesarias para la implementación de la aplicación.

- Selección de *framework*
- Selección de lenguajes de programación
- Prototipo inicial programado que verifique que las tecnologías seleccionadas pueden cumplir con las necesidades, como poder ver las otras aplicaciones instaladas por el usuario.

Sprint 3 Durante el tercer sprint se tomó en cuenta otros aspectos importantes para mejorar la interfaz de usuario.

- Creación de logos
- Establecer el nombre de la aplicación a Ela
- Temporalmente se conectó el asistente virtual a ChatGPT para que los usuarios tengan una visión más completa de cómo se usa la aplicación. No será el modelo final utilizado.

Problemas encontrados:

- ChatGPT se tardaba bastante en responder. Los usuarios no sabían si la aplicación se había trabado o si estaba funcionando. Se solucionó agregando una animación que muestra cuando el asistente está "escribiendo".
- Originalmente no se había usado paralelismo, así que la interfaz se bloqueaba mientras se realizaba la comunicación con ChatGPT. Tampoco se conocía cómo hacer paralelismo en Kotlin. Se solucionó investigando e implementando lo aprendido.

Sprint 4 Durante el cuarto sprint se buscó mejorar aspectos pendientes de la interfaz de usuario. Este sprint fue preparación para empezar aspectos técnicos de la aplicación.

- Diseño de la pantalla de ajustes
- Limpieza de código

Debido a retroalimentación del usuario se agregaron los siguientes cambios adicionales:

- Modo día y modo noche. A los usuarios les molestaba el fondo blanco de la aplicación cuando la usaban durante la noche.
- Un patrón repetitivo como fondo. Mencionaban que se miraba muy vacía la pantalla principal.

Problemas encontrados:

- Una buena parte de este sprint fue utilizado resolviendo problemas al instalar unas librerías.

Sprint 5 Para el quinto sprint se empezó a trabajar con el bloqueo de tráfico red.

- Implementación de la solicitud de permisos necesarios para la aplicación (notificaciones, VPN, etc)
- Cumplir con los requerimientos de Google para aplicaciones que usan VPN:
  - Creación de un *Quick Tile*
  - Notificación persistente
  - Pedir los permisos requeridos hasta el último momento
- Implementación de trabajo en segundo plano
- Creación de la interfaz de VPN

Entre los problemas encontrados este sprint se encontró:

- Los requerimientos para pedir permisos dinámicos fueron bastante complejos.
- La VPN requiere que se pidan dos permisos: notificaciones y usar una VPN. El permiso de usar VPN se debe de pedir después de tener permisos de notificaciones. Este aspecto no estaba escrito en la documentación de Android y causó que se perdiera bastante tiempo del *sprint*.
- Sincronización de datos entre procesos aislados. El proceso usado para *Quick Tile*, trabajo de segundo plano y la UI son creados independientemente pero deben compartir información. Se solucionó usando la base de datos *Data Store*

Sprint 6 El objetivo del sexto sprint fue terminar con todos los detalles técnicos necesarios antes de introducir modelos de inteligencia artificial.

- Implementación de base de datos para los chats y los bloqueos

- Dependiendo de la configuración del usuario, al encender el teléfono Ela se encenderá automáticamente.
- Limpieza de código
- Por retroalimentación de usuario, se agregó la opción de permitir acceso a dominios maliciosos que en realidad no eran maliciosos usando una *whitelist*.
- Nuevas animaciones gráficas

Las dificultades encontradas en este sprint fueron:

- La VPN bloqueaba todo el tráfico. Este problema no se logró solucionar durante este sprint.
  - Se recibían paquetes IP puros pero era necesario solo parsear y reenviar solo los contenidos.
  - Por limitaciones técnicas se necesitaba implementar un intermediario para el tráfico TCP. Esto era mucho más complejo cuando se considera que solo se recibían paquetes IP. No existían librerías que apoyaran.

Sprint 7 Los objetivos del séptimo sprint fueron introducir los modelos de inteligencia artificial y terminar trabajo pendiente del sprint anterior.

- Se cambió la manera en la que se trataba con la VPN para solo filtrar el tráfico DNS y permitir filtrar selectivamente el tráfico a dominios sospechosos.
- Integración de los modelos de Inteligencia Artificial[46].
  - Asistente de NLP
  - Determinar aplicaciones sospechosas basado en los permisos solicitados

Las dificultadas encontradas en el séptimo sprint fueron:

- Problemas con las librerías utilizadas para integrar los modelos de inteligencia artificial
- No se esperaba que el desarrollo del filtrado de tráfico red tomara tanto tiempo de desarrollo.

Sprint 8 Durante el octavo sprint se buscó perfeccionar la aplicación con detalles mostrados en pruebas preliminares.

- Cuando se bloquee un dominio saldrá una notificación flotante para hablar con el asistente.
- Cambiar el modelo de inteligencia artificial:

    - Agregar consultas a servidores *whois* para darle más datos al modelo.
- Añadir la funcionalidad de exportar datos para facilitar el futuro análisis de resultados.
- Mejorar la velocidad de la aplicación con paralelismo en la VPN y la UI.
- Añadir funcionalidad para reiniciar la VPN al terminar de actualizar la aplicación.

  Debido a retroalimentación de los usuarios se realizaron los siguientes cambios adicionales:

    - Se cambió el diseño de la pantalla principal
    - Modificar la paleta de colores
    - Cambiar el icono para reflejar la nueva paleta de colores
    - Pruebas tempranas mostraron que los usuarios solo dejaban la aplicación de fondo y no la usaban. Se solucionó mostrando un consejo de ciberseguridad diario en una notificación.
- Pruebas a largo plazo con los usuarios

Dificultades adicionales encontradas:

- Un grupo de usuarios de prueba encontraron un error poco común que solo ocurría al usar IPv6, especialmente al usar redes móviles. Se solucionó agregando soporte para IPv6.
- Existía un error donde el consejo diario no era mostrado.

Sprint 9 Para el sprint 9 se tenía como meta realizar pruebas de cómo reaccionan los usuarios a un ataque de *phishing* simulado.

- Pruebas con los usuarios
- Realizar un ataque de *phishing* simulado (pruebas preliminarkj)

7.3. Pruebas de usabilidad

7.3.1. Pruebas de corto plazo

Las pruebas de usabilidad fueron vitales para el desarrollo de la interfaz. Permitieron un ciclo rápido de retroalimentación. Esto fue especialmente importante en las partes iniciales de la aplicación.

Las pruebas de corto plazo generalmente fueron realizadas al final de los *sprints* donde se realizaron cambios visuales a la aplicación; específicamente en los *sprints* 1, 3, 4, 6, y 8.

Asimismo, cada prueba fue realizada por dos o tres participantes. Los participantes comúnmente se mantenían en el rango de edades de 20 a 25 años; pero en los últimos *sprints* (6 y 8) se buscó diversificar y se empezó a trabajar con el rango de edad de 40-50 años.

Entre los cambios originados por retroalimentación se destaca:

- Cambio del icono de guardar
- Modo oscuro
- Elección de una de colores alegre
- Agregar un fondo
- Agregar una lista de dominios que no serán bloqueados
- Agregar la opción de bloquear ningún tráfico, únicamente advertir cuando se encuentre tráfico a un dominio malicioso.

Cabe destacar que no toda la retroalimentación mencionó explícitamente qué problema existía en la aplicación; en la mayoría de casos se encontró el problema en base a la observación.

Por ejemplo, en una de las primeras pruebas de usabilidad, cuando los usuarios abrían la aplicación en la noche, ellos fruncían el ceño y entrecerraban los ojos por un momento. De manera que se decidió implementar un modo oscuro para que no molestara a los usuarios cuando usaran sus teléfonos en condiciones de baja luz.

7.3.2. Pruebas de largo plazo

Por otro lado, las pruebas de largo plazo únicamente fueron realizadas cuando toda la funcionalidad de la aplicación estaba completa o casi completa. De manera que esta prueba fue desarrollada a la mitad del *sprint* 8 y en el *sprint* 9.

Se contaron con seis participantes para la prueba de largo plazo. Se contaba con 3 hombres con las edades de 16, 23 y 52. Así como cuatro mujeres de 22 y 23 años.

Las pruebas a largo plazo fueron más efectivas para detectar detalles:

- Ideas de animaciones
- Cambiar los colores a unos más sutiles
- Agregarle color al logo que usualmente se muestra en las notificaciones.
- Módulo de consejo diario de ciberseguridad.

7.4. Módulos

7.4.1. Módulo de bloqueo de dominios maliciosos

Para que la aplicación pueda bloquear tráfico de dominios maliciosos es necesario que pueda examinar todo el tráfico de red del dispositivo. Esto implica que la aplicación puede ser catalogada como un tipo de *Firewall*.

Esto impondrá un desafío ya que Android no permite la instalación de *Firewalls*. La única manera de pasar por esta limitación es obtener permisos administrador. El proceso para obtener permisos de administrador es no intuitivo e incluso puede traer consigo consecuencias negativas como anular la garantía. No es conveniente ni factible que el usuario instale un *Firewall*.

Por otro lado, existe otro tipo de aplicación que cumple los requerimientos de poder supervisar todo el tráfico red e incluso ignorar dinámicamente algunos paquetes red, las aplicaciones VPN. A pesar de que no fueron diseñadas con esa funcionalidad en mente, las aplicaciones VPN interceptan todo el tráfico de red y lo redirigen a la red destino. Finalmente, la instalación de una aplicación VPN no requiere permisos de administrador para ser instalada.

A raíz de todo lo anterior, se decidió que la aplicación bloqueará el tráfico de red de manera similar a una aplicación *Firewall*, pero fuera identificada como una VPN por el dispositivo.

Otra parte importante para la filtración de tráfico red es clasificar qué dominios son maliciosos. Esto fue a través de una red neuronal multicapa[46] que recibe información como la longitud del dominio, cantidad de números, fecha en la que fue registrado, etcétera. La mayoría de la información se pudo recolectar directamente desde el dominio, pero para algunos aspectos fue necesario realizar consultas a servidores Whois.

Por otro lado, Android SDK provee las APIs necesarias para implementar una aplicación de VPN. No fue necesario añadir más tecnologías al proyecto.

Convenientemente, las APIs provistas para VPN únicamente interceptan el tráfico red de salida del dispositivo. Es decir, el tráfico interceptado no será enviado automáticamente a ningún lugar. Esta libertad le permite a la aplicación enviar el tráfico filtrado directamente a internet, sin necesidad de utilizar un servidor VPN.

Otra ventaja de las APIs de Android SDK para aplicaciones VPN, es que permiten cierto nivel de libertad sobre el tipo de conexión que será creada al conectarse a la VPN. Manualmente se puede modificar la tabla de ruteo del teléfono para que solo cierto tráfico vaya dirigido a la VPN.

Sin embargo, una desventaja significativa fue que el API para VPN interceptaba paquetes de la capa de enlace. En otras palabras, se reciben paquetes IP y se espera paquetes IP como respuesta al tráfico interno. Por si fuera poco, Android SDK únicamente provee APIs para enviar información a través de la red usando el protocolo TCP y UDP.

Esto significa que para reenviar un paquete a internet era necesario extraer el contenido

de los paquetes IP, detectar si se trataba de una conexión TCP o UDP y reenviarlo a internet. Esto aumenta considerablemente la manera de trabajar con TCP; ya que sería necesario implementar un *three-way-handshake* solo para extraer el contenido que se desea reenviar a internet.

Tampoco existen librerías de Java o Kotlin que implementen el algoritmo TCP. Después de todo, generalmente el protocolo TCP se encuentra implementado directamente desde el sistema operativo.

Con toda la información anterior, es necesario encontrar una manera efectiva de diseñar la manera en la que se filtrará el tráfico a dominios maliciosos. Para esto, a lo largo del proyecto se cambió la manera en la que se estaba bloqueando el tráfico de red:

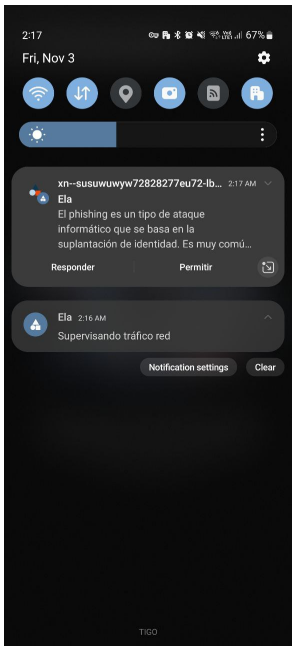


Figura 6: Notificación al haber bloqueado un dominio malicioso

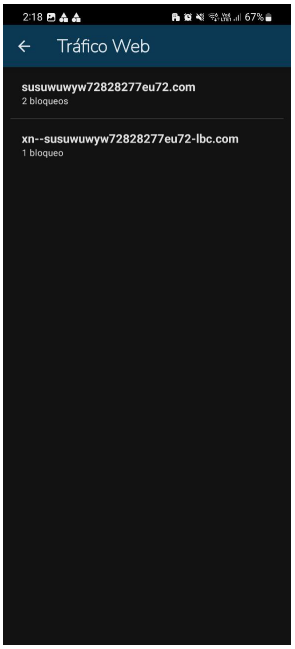


Figura 7: Pantalla de ver detalles sobre los dominios bloqueados hoy

Primera acercamiento Originalmente se tenía planeado que todo el tráfico del dispositivo fuera enviado a la VPN, donde sería analizado para ser reenviado al internet. Específicamente, se centró en analizar el tráfico creado por el protocolo HTTP y HTTPS. Se eligió este protocolo ya que es uno de los protocolos más comunes para transmitir información en internet.

Este acercamiento contaba con dos desventajas importantes:

1. HTTPS y HTTP utilizan TCP, lo que lo vuelve complicado de manejar en estas circunstancias.

2. El protocolo HTTPS maneja encriptación. De manera que para cuando llega a la capa de enlace de datos, es imposible reconstruir el paquete. No se puede conseguir el nombre de dominio con el que se está comunicando.

En otras palabras, este acercamiento tenía una implementación compleja y no permitiría conseguir la información que realmente se necesita para que la aplicación decida si debe bloquear el tráfico.

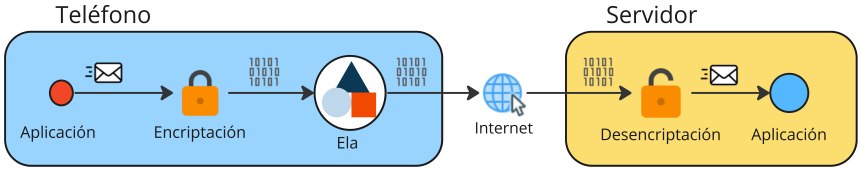


Figura 8: Primer acercamiento para el bloqueo de tráfico red

Segundo acercamiento Dada la imposibilidad de desencriptar la información transmitida por HTTPS se buscó un nuevo acercamiento. La aplicación podría hacerse pasar por el servidor para obtener la clave privada para encriptación; de manera que sería capaz de desencriptar el tráfico saliente del teléfono. Después de leer esta información, se volverá a crear una conexión HTTPS y enviará la información extraída.

Este acercamiento parecería solucionar el problema de la encriptación del primer acercamiento, pero tiene sus propias dificultades. Debido a que no se cuentan con los certificados de encriptación adecuados, se detectará inmediatamente que no se está comunicando directamente con el servidor. Esta solución traía consigo el riesgo de que todo el tráfico HTTPS fuera bloqueado cuando se detectara que no se estaba comunicando con el servidor correcto.

Debido a lo anterior, se consideró que este acercamiento no fue factible y se buscó otra alternativa.

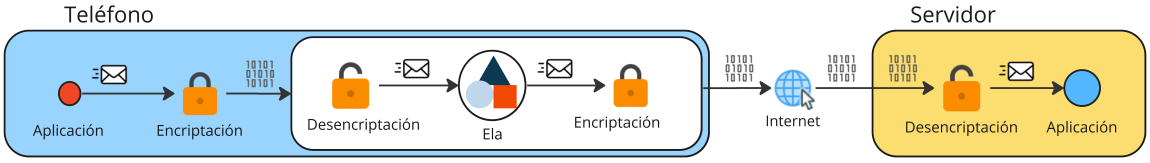


Figura 9: Segundo acercamiento para el bloqueo de tráfico red

Tercer acercamiento La tercera alternativa decide bloquear el protocolo DNS en vez del protocolo HTTPS. En vez de bloquear los mensajes a los dominios maliciosos, se hará que no se pueda encontrar cuáles son sus direcciones IP relacionadas; efectivamente volviendo la comunicación hacia estos imposible.

Esto trae consigo muchas ventajas:

- El protocolo DNS no está cifrado por defecto
- A diferencia de HTTPS, DNS utiliza el protocolo UDP. Volviéndolo significativamente más fácil de procesar.
- Al bloquear únicamente la traducción de direcciones IP, puede bloquear indirectamente otros tipos de tráfico de red.

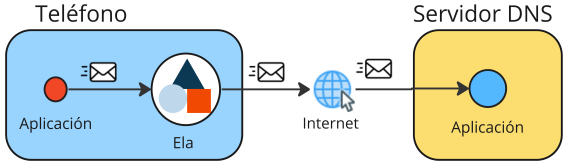


Figura 10: Tercer acercamiento para el bloqueo de tráfico red

7.4.2. Módulo de aplicaciones sospechosas

Para el desarrollo del módulo sospechosas se utilizó un modelo de inteligencia artificial[46]. Este modelo puede ser clasificado como un modelo de red neuronal multicapa. Para utilizarlo se le debe enviar únicamente una lista sobre los permisos que tiene una aplicación específica.

Cabe mencionar que originalmente este módulo era llamado aplicaciones peligrosas, pero fue renombrado a aplicaciones sospechosas. Durante las últimas pruebas de corto plazo, se determinó que, los usuarios pensaron que si una aplicación era juzgada como peligrosa, entonces era maliciosa; aspecto que no era cierto.

Esto causó que cuando los usuarios de la primera prueba larga vieran aplicaciones confiables (como Duolingo) catalogadas como peligrosas tuvieran dos reacciones: Los usuarios se sentían confundidos y un poco temerosos de que estas aplicaciones tuvieran *malware*. Por otro lado, los usuarios también mencionaron que simplemente perdieron confianza en Ela al pensar que estas aplicaciones fueron catalogadas como *malware*.

Al momento de realizar la prueba de largo plazo, ya se le había cambiado de nombre al módulo. De manera que los usuarios interpretaron correctamente que sospechosa no necesariamente significaba que la aplicación era maliciosa. Esto provocó que todos los usuarios reaccionaran de manera diferente; todos indicaron sentirse curiosos por los permisos que utilizan esas aplicaciones.

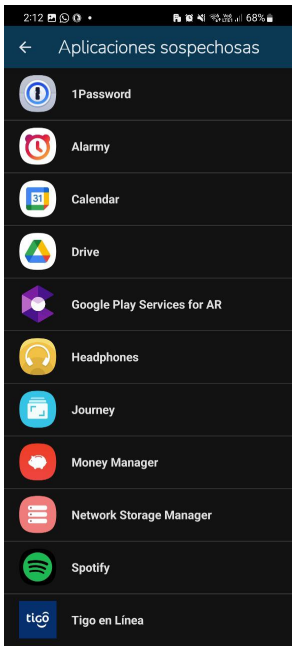


Figura 11: Pantalla de aplicaciones sospechosas

7.4.3. Módulo de asistente virtual

La aplicación incluyó un módulo de un asistente virtual al que se le puede preguntar sobre a ciberseguridad. Al inicio, para de que los usuarios probaran el funcionamiento de manera temprana, se utilizó el API de ChatGPT. Sin embargo, durante las últimas etapas de desarrollo se migró a un modelo especializado responder dudas sobre ciberseguridad[46].

Todos los usuarios de la primera y segunda prueba de usabilidad admitieron que interactuaban principalmente con la aplicación a través del módulo del asistente virtual. Sin embargo,

Al final de la prueba de usabilidad de largo plazo, los seis usuarios admitieron que interactuaban principalmente con la aplicación a través del módulo del asistente virtual. Sin embargo, un 66 % de ellos mencionó que después de cierto tiempo, ya no tenían más dudas para preguntarles al asistente virtual. De manera que ya no interactuaban tan seguido con el agente.

Esto representó un desafío, ya que se deseaba que los usuarios siguieran interactuando y aprendiendo de ciberseguridad con el asistente. De manera que se decidió complementar este módulo con el módulo del dato curioso de ciberseguridad.

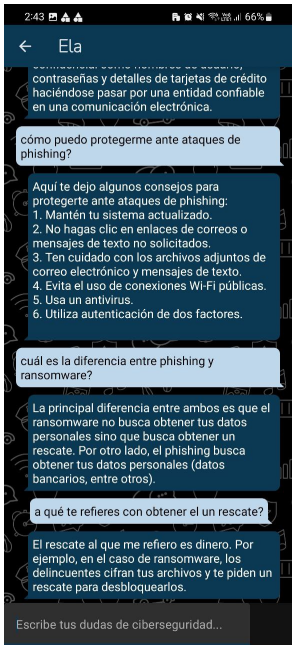


Figura 12: Pantalla de solicitarle una pregunta a Ela

7.4.4. Módulo de dato curioso de ciberseguridad

Para buscar que los usuarios interactuaran más con la aplicación, se buscó mostrarle al usuario un dato curioso sobre seguridad. De esta manera.

Cada día el usuario recibiría dos notificaciones silenciosas. Las notificaciones se mos-

trarían aproximadamente a las 8 am y a las 8 pm. Ambas contendrían un dato curioso de ciberseguridad, generados por el asistente virtual. De manera que el usuario tendría nueva información que podría preguntarle al agente. Por otro lado, aún si el usuario decidiera no volver a interactuar con el asistente, todavía podrían aprender de ciberseguridad a través de estos datos curiosos.

Cinco días después de introducir el módulo de dato curioso, el 33 % de los usuarios de la prueba a largo plazo mencionaron que aumentó su interacción con el asistente virtual. Asimismo, el 83 % de los participantes indicó que les parecía interesante los datos que se les presentaba diariamente.

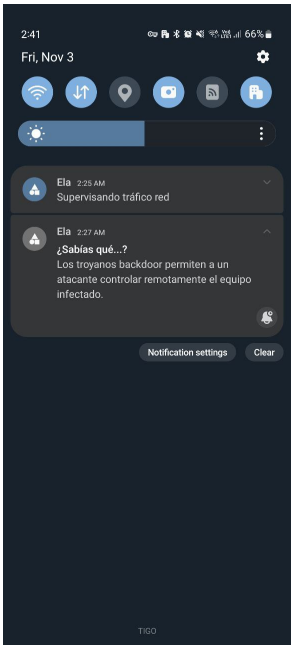


Figura 13: Notificación del consejo diario

7.5. Prueba preliminar de aprendizaje

A fines de probar la efectividad de Ela, se realizó una prueba preliminar con un pequeño grupo de cuatro personas. De ellas, una persona era un adolescente, dos personas eran adultos jóvenes y la última persona era un adulto. Cabe destacar que todos los participantes habían utilizado Ela por al menos dos semanas. Asimismo, todos los integrantes aceptaron voluntariamente a participar en pruebas de naturaleza sensible y sus implicaciones; pero no sabían con exactitud qué tipo de prueba se iba a realizar ni el momento en el que se pasarían.

A diferencia de la prueba de aprendizaje final, la prueba preliminar se centró únicamente en determinar si Ela podía ayudar a un usuario a defenderse de un ataque simulado. Para esto, se decidió diseñar un ataque de *Spear Phishing*.

Al ser una prueba preliminar, todos los usuarios contaban con la versión completa de la aplicación. Es decir, cada vez que la aplicación detectaba y bloqueaba tráfico a dominios

maliciosos, procedería a darle un poco de información relevante al usuario sobre este tipo de ataque.

Una vez se decidió el tipo de ataque que se iba a simular, se prosiguió a seleccionar el vector de ataque. Entre los vectores de ataque se consideró:

- Mensaje de texto
- Correo electrónico
- Llamada telefónica

De estas, se decidió utilizar correo electrónico. Esto fue debido a la dificultad de obtener un nuevo número telefónico nuevo. Los participantes conocían los números telefónicos del equipo y no se quería involucrar a terceros en el ataque para mantener la confidencialidad.

Considerando que se decidió usar correo electrónico como vector de ataque, era necesario decidir de qué iba a tratar el *phishing*. Después de una lluvia de ideas, se llegó a las siguientes opciones:

- Cambio de contraseña en una cuenta social
- Nuevo inicio de sesión en alguna cuenta que tenga el usuario
- Un correo que mencionara que su medio de pago había sido rechazado.

Dado a que se deseaba utilizar el mismo vector de ataque entre todos los participantes y que uno de ellos era un adolescente, se descartó la tercera idea. La primera y segunda idea eran relativamente similares, pero se decidió por un nuevo inicio de sesión. La primera opción contaba con el problema de que comúnmente cuando se cambia una contraseña se cierra sesión en todos los dispositivos y era imposible de saber si el participante se encontraba usando activamente usando la plataforma. Si este hubiera sido el caso, el usuario podría haber empezado a sospechar por el ataque simulado.

Finalmente, considerando que se tenía el correo de todos los participantes y no se sabía con exactitud qué redes sociales utilizaban, se decidió que el vector de ataque final fuera un nuevo inicio de sesión de su cuenta de correo.

Una vez se decidió el método de ataque, se enfrentó a un desafío que no había sido considerado. Gmail tiene un buen sistema para detectar *phishing* relacionado a Google. Por lo que el ataque simulado fue detectado por Gmail en minutos y la cuenta utilizada para mandar correos fue inmediatamente marcada como maliciosa.

Tomando en cuenta los descubrimientos anteriores, se prosiguió a cambiar el vector de ataque a un *phishing* sobre Facebook. El nuevo ataque consistía en un correo mencionando que hubo un nuevo inicio de sesión (Figura 14). Si el usuario seleccionaba el botón de verificar actividad, serían redirigidos a una página falsa de inicio de sesión (Figura 15). Posteriormente, si le daban clic en el botón de iniciar sesión serían reenviados a una página con el mensaje de que habían caído en *phishing*. Es muy importante mencionar la página de inicio de sesión falsa no guardaba ni transmitía información relacionada a sus credenciales.

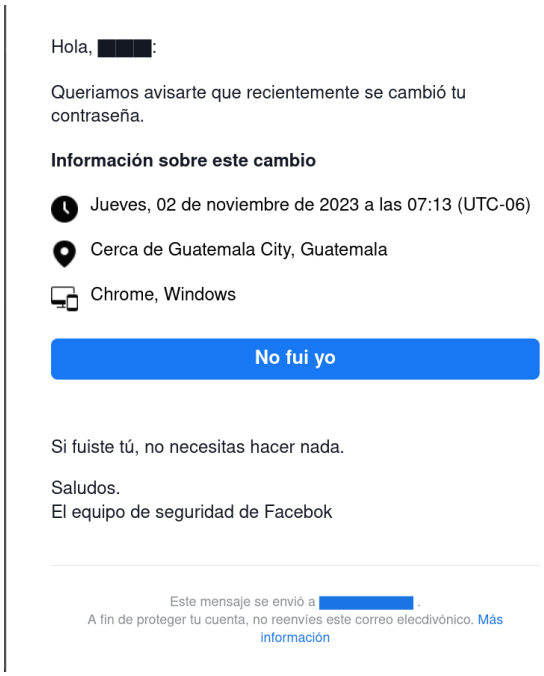


Figura 14: Correo de *phishing* para la prueba preliminar

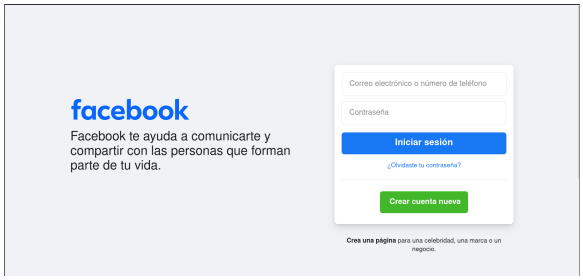


Figura 15: Página de inicio de sesión falsa para la prueba preliminar

Figura 16: Página mostrada cuando tratan de iniciar sesión

Adicionalmente, con fines de darles un par de pistas a los participantes para identificar al correo de *phishing*, intencionalmente se añadieron algunas faltas ortográficas sutiles al correo de *phishing*.

Resultados de la prueba preliminar

Cada participante reaccionó de manera diferente al ataque simulado; pero ninguno ingresó sus credenciales en la página falsa.

La persona adolescente decidió abrir el correo e inmediatamente detectó que se trataba de un correo de *phishing*. Después de esta realización, no volvió a interactuar con el correo.

La persona adulta no detectó el ataque de manera inmediata. Le dio clic al botón de "no fui yo". Cuando Ela bloqueó la página web, sospechó correctamente que el correo era una prueba realizada como parte de este estudio. A pesar de que en ese momento no tenía ninguna prueba de que era un *phishing* inofensivo, decidió apagar temporalmente la protección de Ela y ver el sitio falso. No ingresó sus credenciales en el sitio falso.

Una de los dos adultos jóvenes ignoró completamente el correo de *phishing*. Al no interactuar con el correo, se defendió perfectamente contra el ataque.

Por último, el segundo adulto joven presentó un caso bastante particular: en el pasado le habían robado una cuenta de Instagram. De manera que al ver el correo de un inicio de sesión desconocido cayó el pánico. Intentó darle clic al botón de "no fui yo", pero Ela fue capaz de bloquear el sitio web. Al ver la notificación explicando que el sitio era un ataque de *phishing*, el participante decidió preguntarle con más detalle cómo identificar un ataque de *phishing*. Siguiendo las recomendaciones de Ela, se dio cuenta que el correo no fue enviado por una cuenta de Facebook y otros detalles relevantes. Al estar seguro de que era un ataque de *phishing*, inició sesión en el sitio verdadero de Facebook a través de computadora para cambiar su contraseña.

7.6. Prueba piloto de aprendizaje

La prueba piloto de aprendizaje fue realizada con un grupo de 12 personas. De ellos, nueve estaban en el rango de 18 a 25 años (Figura 17); de manera que se trabajó principalmente

con adultos jóvenes. Adicionalmente, la mitad de estos participantes fueron asignados como parte del grupo de control.

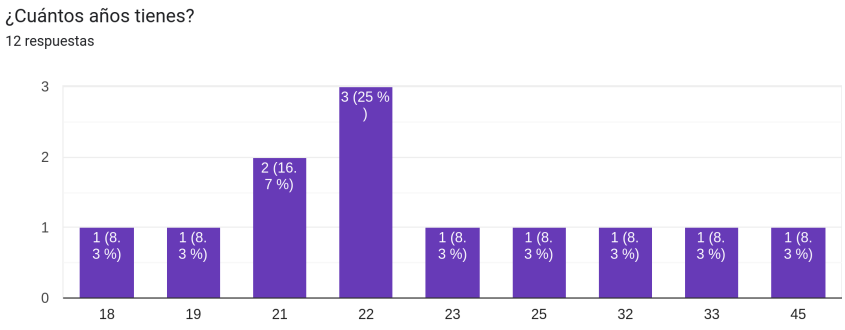


Figura 17: Edades de los participantes en las pruebas de aprendizaje

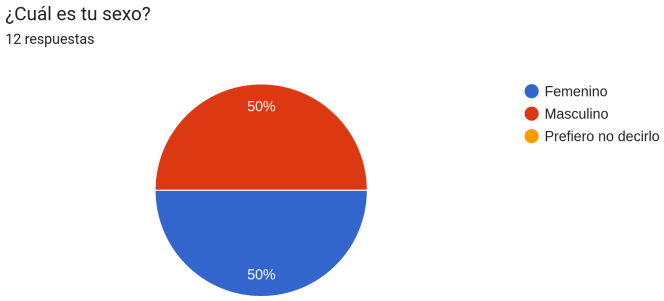


Figura 18: Sexo de los participantes en las pruebas piloto de aprendizaje

Cabe mencionar que todas las personas habían utilizado la aplicación Ela por al menos una semana. De manera que todos estaban familiarizados con el uso de la aplicación.

7.6.1. Ataque simulado

Al igual que en las pruebas preliminares, se decidió simular un ataque de *Spear Phishing*. Sin embargo, a diferencia de la prueba preliminar, se decidió usar dos vectores de ataque.

Correo de *Spotify*

El primer vector de ataque fue un mensaje de correo de Spotify mencionando que su método de pago había sido rechazado y su plan *premium* iba a expirar próximamente (Figura 19).

Se decidió imitar a Spotify ya que se había determinado que todos los integrantes contaban con una cuenta de Spotify y un 75 % de ellos mencionó que lo usaban de manera frecuente. Se esperaba que los usuarios con *premium* se preocuparan sobre su método de

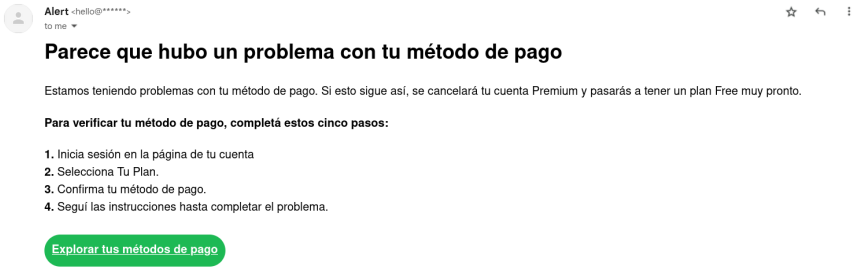


Figura 19: Correo enviado para la prueba de aprendizaje

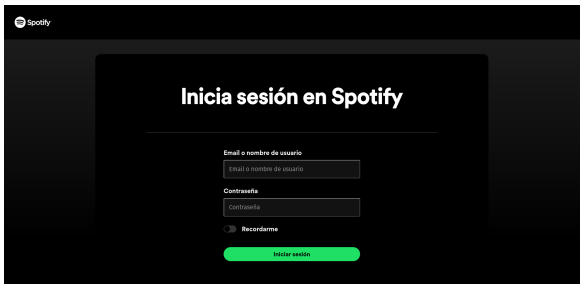


Figura 20: Página de inicio de sesión falsa para la prueba de aprendizaje

pago e iniciaran sesión; mientras que los usuarios que no tuvieran *premium* se sintieran confundidos y curiosos e iniciaran sesión. Asimismo, se volvió a decidir usar electrónico por el éxito demostrado en los ataques de la prueba preliminar.

El correo incluiría un enlace a una página de inicio de sesión falsa, donde se le pediría su usuario y contraseña (Figura 20). En caso el usuario ingresara sus credenciales, el usuario sería redireccionado a una página donde se le informaría que había caído en *phishing*.

Sin embargo, hubo un impedimento importante: los correos fueron enviados a la carpeta de *Spam* de los participantes. Esto provocó que ninguno de los participantes llegaran a ver el correo de *pshishing*. De manera que se optó por usar un segundo ataque.

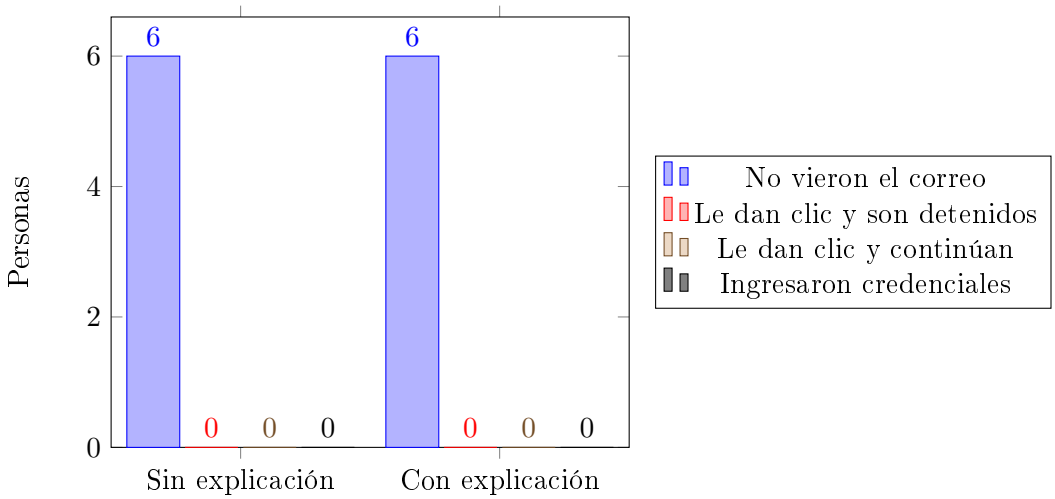


Figura 21: Resultados de las pruebas de *phishing* de *Spotify*

Mensaje SMS

Para evitar que el segundo ataque simulado volviera a ser enviado a la carpeta de Spam, se decidió que el ataque simulado no fuera a través de correo electrónico.

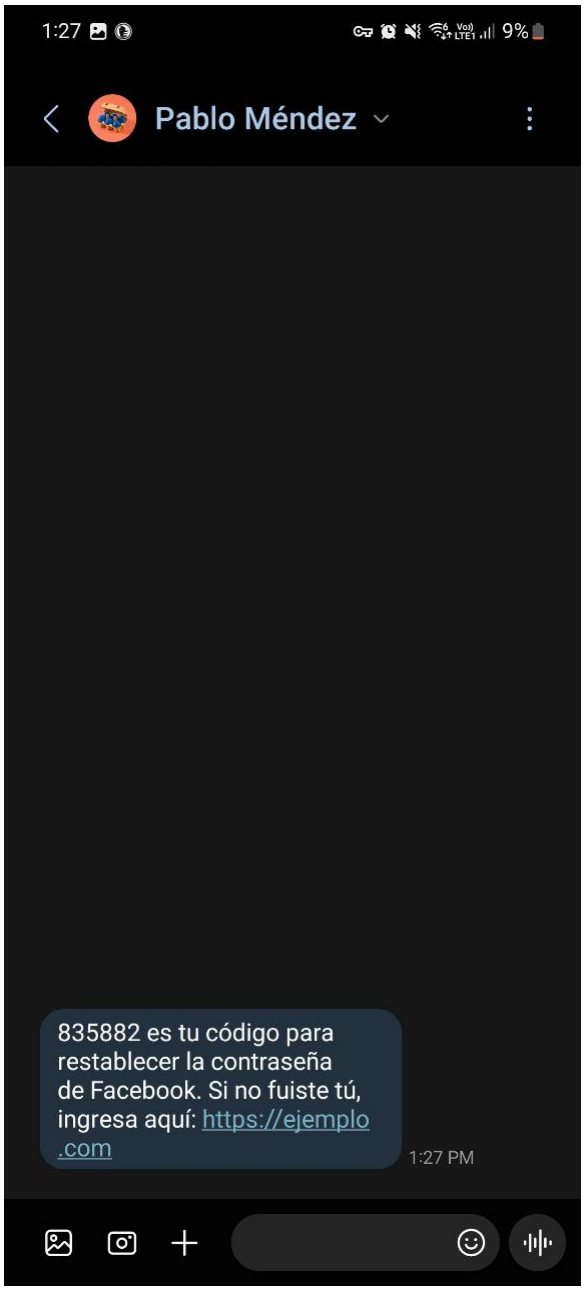


Figura 22: Mensaje de texto sobre un *phishing* de Facebook

Dado lo anterior, se decidieron tres nuevos métodos de ataque:

- SMS (mensaje de texto)
- Llamada telefónica

- Mensaje en redes sociales

Inmediatamente se descartó la opción de realizar una llamada telefónica. Los participantes ya conocían a los miembros del equipo, de manera que podrían reconocer su voz. Aunque hubiera sido posible utilizar un *software* de voz artificial, se temía que los participantes detectaran la anomalía en la llamada.

Cabe mencionar que se deseaba utilizar exactamente el mismo ataque entre todos los participantes. Esto impidió que se usara un mensaje en redes sociales. Cada usuario utilizaba diferente redes sociales y algunos directamente bloqueaban los mensajes de aquellos que no estuvieran en sus contactos en estas plataformas.

Dado lo anterior, se decidió que segundo ataque simulado sería a través de un SMS. Esto presentaba un problema, los participantes conocían el número telefónico de los miembros del equipo. Para solucionar esto, se consiguió un nuevo número telefónico para uso específico de esta prueba.

Una vez decidido lo anterior, se debía determinar el contenido del SMS. Entre las alternativas consideradas por el equipo estaba:

- Actividad maliciosa en Facebook
- Ganaron un torneo
- Su compañía de telecomunicaciones les emitió una nueva factura

Mientras que ninguna alternativa contaba con desventajas significativas, se decidió optar por la primera. Ya se había demostrado durante el estudio preliminar que hacerse pasar por Facebook sí era capaz de engañar a los participantes.

Resultados del ataque SMS El 33 % de los usuarios que tenían la versión sin explicación y el 50 % de los usuarios con explicación ignoraron completamente el mensaje de *phishing* (Figura 23). A raíz de esto, Ela nunca necesitó advertirles sobre el tipo de *malware* al que se arriesgaban por hacerle clic al enlace en el mensaje.

Por otro lado, los integrantes que no ignoraron el mensaje procedieron a hacerle clic al enlace malicioso en el SMS. De ser el caso, Ela bloquería el acceso al enlace y le mostraría una alerta mencionando que la página fue bloqueada. Si el usuario tenía la versión con explicación de Ela, también se le mostraría un resumen de qué es el *phishing* y cómo podría protegerse de él. A partir de esto, el participante tendría dos opciones, dejar de tratar de acceder al enlace o ignorar la advertencia y seguir tratando de acceder al enlace.

De los integrantes que le dieron clic al enlace malicioso, el 100 % de los integrantes con la versión con explicación y el 50 % de los usuarios con la versión sin explicación dejaron de tratar de acceder a la página. Esto es de vital importancia porque demuestra que brindar detalles sobre el tipo de dominio malicioso aumenta en un 50 % la probabilidad de que una persona se defienda exitósamente ante un ataque de este tipo.

Finalmente, de los usuarios con la versión que ignoraron la advertencia, el 50 % ingreso sus credenciales en la página de *phishing*, siendo víctimas del ataque.

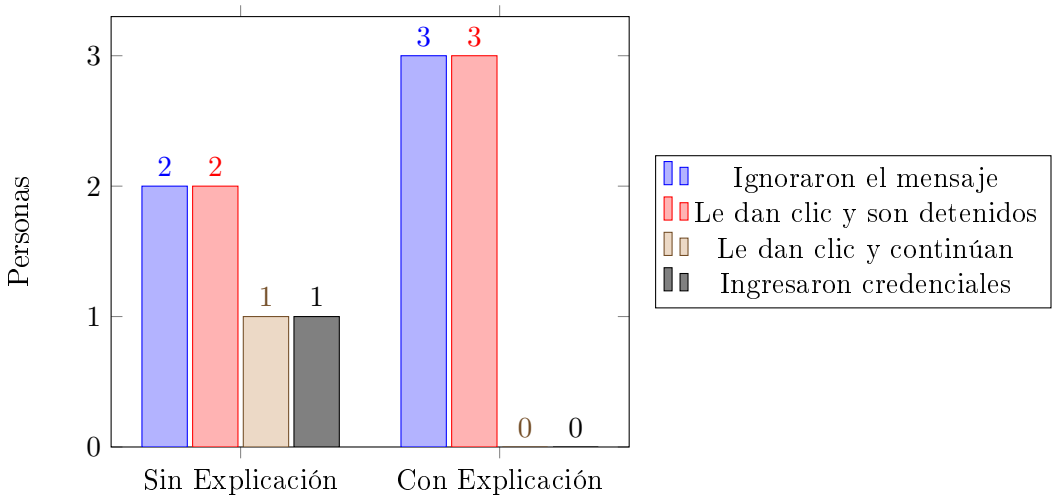


Figura 23: Resultados de las pruebas de *phishing*

7.6.2. Conocimiento de ciberseguridad

Antes del uso de Ela el 75 % de los sujetos de prueba no lograba identificar el *ransomware* de otros tipos comunes de *malware* (Figura 24). Asimismo, solo un 42 % admitieron que no tenían el conocimiento suficiente para determinar qué era el *ransomware*.

Sin embargo, después de haber utilizado Ela por más de 10 días, se puede observar una diferencia clara sobre sus conocimientos sobre este tipo de virus. Al finalizar la prueba, el 58 % de los participantes identificó correctamente un caso de *ransomware*.

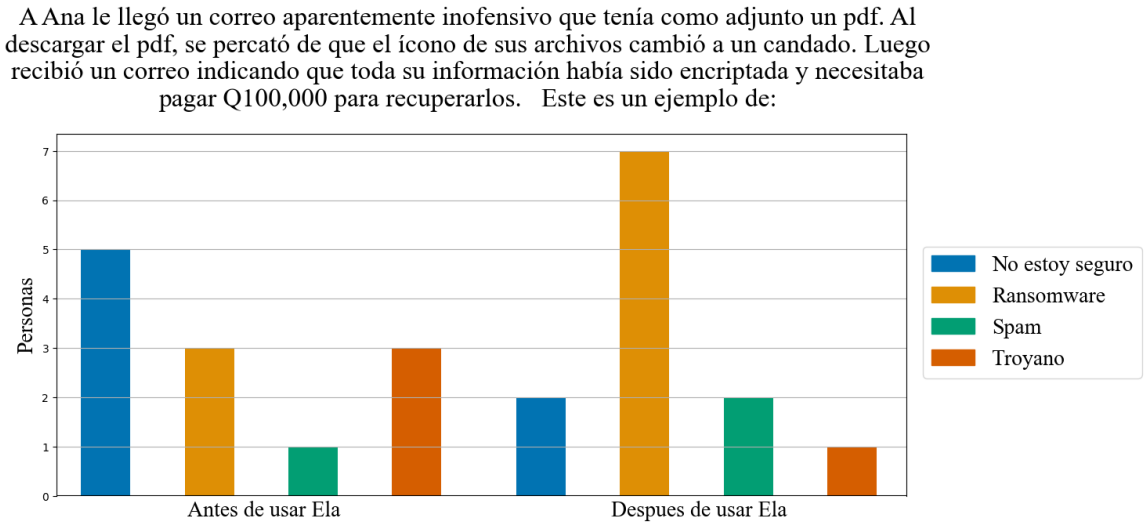


Figura 24: Evolución de las pregunta de conocimientos de ciberseguridad sobre *malware* antes y después de haber usado Ela

De la misma manera, se observó si Ela afectaba en la manera que los participantes reaccionaban hacia ataques de *phishing* potenciales (Figura 25). No hubo diferencia en la manera que los participantes mencionan que reaccionarían. Sin embargo, esto pudo ser causado por-

que la mayoría de participantes (91 %) identificaron desde el inicio la reacción correcta; de manera que el uso de Ela pudo solo haberles reforzado su visión. También es importante reconocer que la manera en la que los usuarios dicen que acturían no necesariamente refleja la manera en la que actúan en la vida real. Como se pudo observar en la Figura 23, muchos participantes todavía caen en ataques de *phishing*.

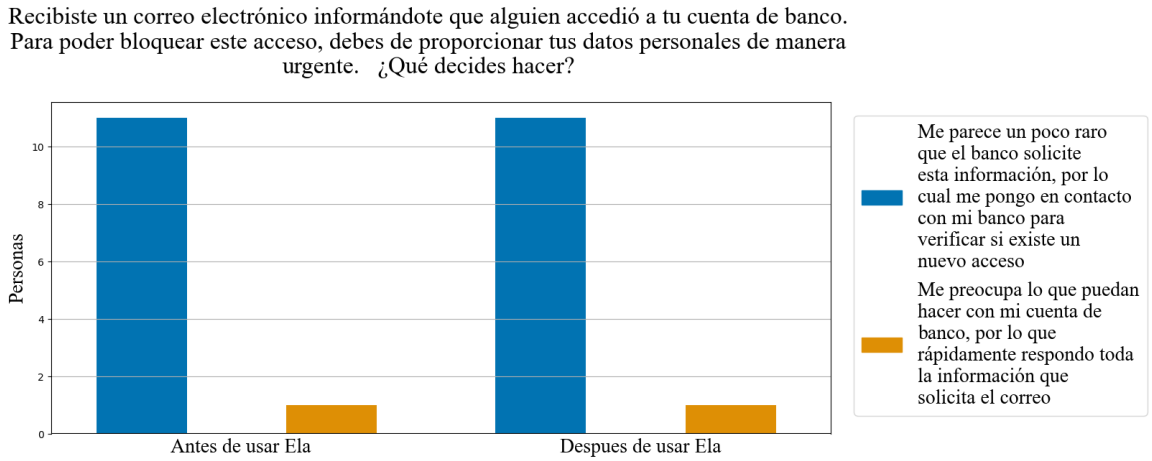


Figura 25: Evolución de las pregunta de conocimientos de ciberseguridad sobre reacciones ante el *phishing*, antes y después de haber usado Ela

Asimismo, antes de usar Ela, 33 % de los participantes no sabían qué era la autenticación de dos pasos (Figura 26). Esto fue particularmente preocupante; el segundo factor de autenticación es un método efectivo para evitar el robo de cuentas. No obstante, después de utilizar Ela, el 83 % de los participantes identificó correctamente qué era autenticación de dos pasos. Al mismo tiempo, el 16 % restante reconoció correctamente que no sabía que era el segundo factor de autenticación.

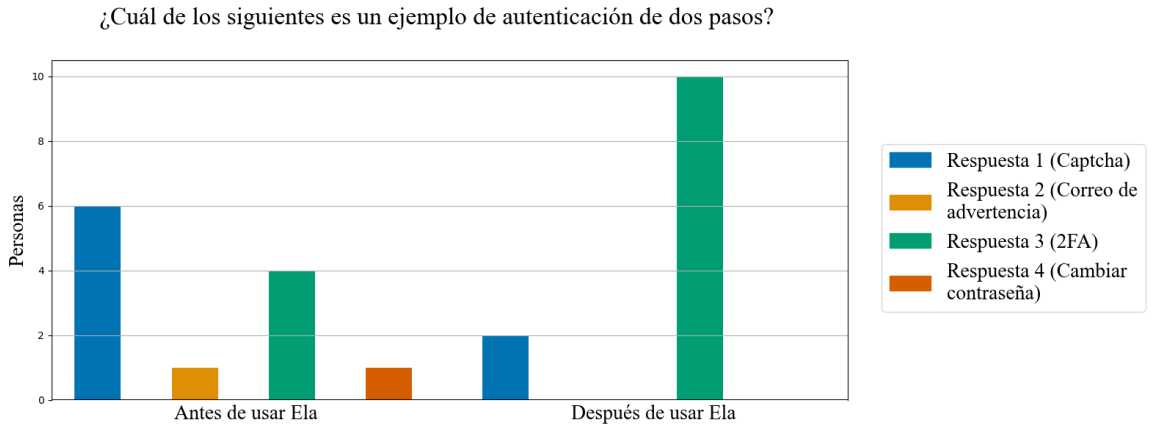


Figura 26: Evolución de las pregunta de conocimientos de ciberseguridad sobre autenticación multifactor antes y después de haber usado Ela

De la misma manera, a través del uso de Ela, la cantidad de participantes que sabían qué es el *vishing* aumentó del 8 % al 75 % (Figura 27). Sin embargo, cabe destacar que no todos los resultados fueron positivos. Por ejemplo, se incrementó en 8 % la cantidad de participantes que pensaron que el *vishing* era una técnica de cifrado de datos.

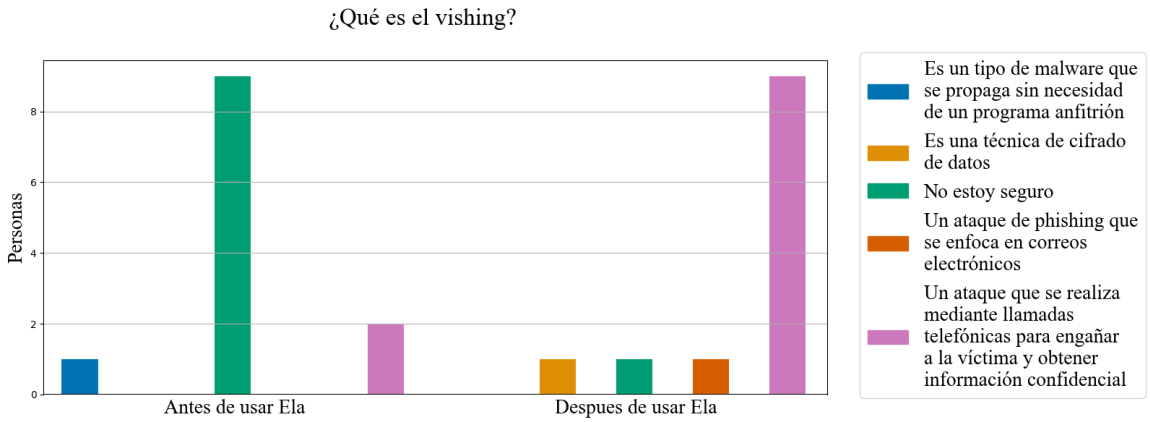


Figura 27: Evolución de las pregunta de conocimientos de ciberseguridad sobre otros tipos de *malware* antes y después de haber usado Ela

Uno de los resultados más interesantes fue al preguntarle a los usuarios que identificaran qué era la ingeniería social (Figura 28). Aunque después de usar Ela se aumentó en 8 % la cantidad de usuarios que identificaban correctamente la ingeniería social; el 83 % seguía teniendo un concepto incorrecto sobre el tema. Esto refleja que, aunque Ela puede mejorar conocimientos de ciberseguridad, no tendrá la misma efectividad en todas las áreas.

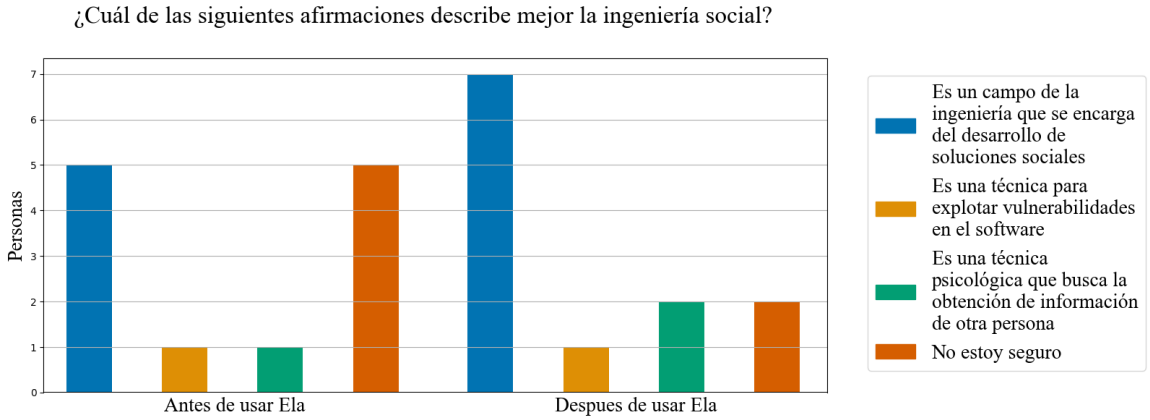


Figura 28: Evolución de las pregunta de conocimientos de ciberseguridad sobre ingeniería social antes y después de haber usado Ela

Otro aspecto que se notó al realizar las pruebas fue que el uso de la aplicación podría causar que el usuario se sintiera más inseguro sobre su conocimiento de ciberseguridad. Esto se evidencia claramente al preguntarles sobre qué harían si creen que tienen un virus (Figura 29). Se mostró un incremento del 16 % en los participantes que mencionaron que no estaban seguros de qué harían en esa situación.

Finalmente, utilizar la aplicación de Ela afectó negativamente la manera en la cual los usuarios consideraban que se deberían manejar las copias de seguridad (Figura 30). Se disminuyó por 8 % los participantes que consideraron que era esencial que se cifraran las copias de seguridad.

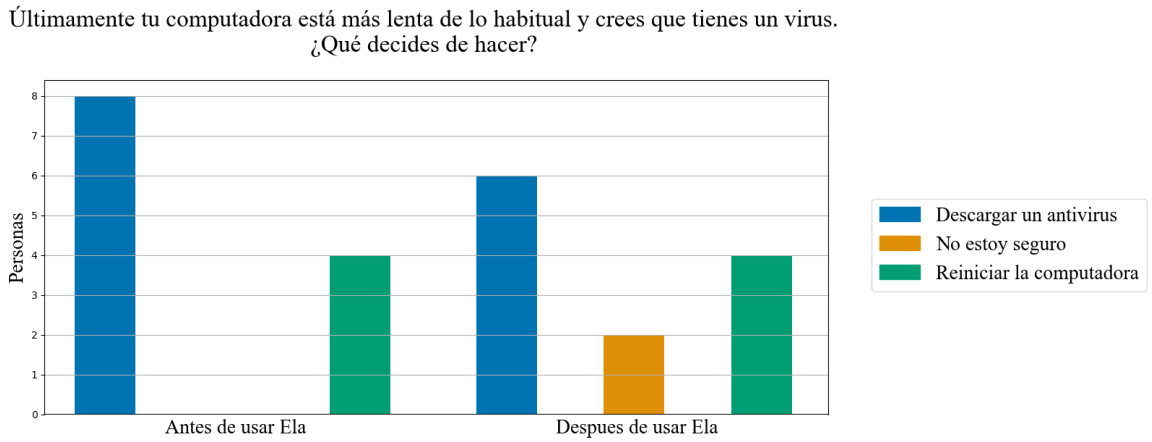


Figura 29: Evolución de las pregunta de conocimientos de ciberseguridad sobre reacciones ante el *malware* antes y después de haber usado Ela

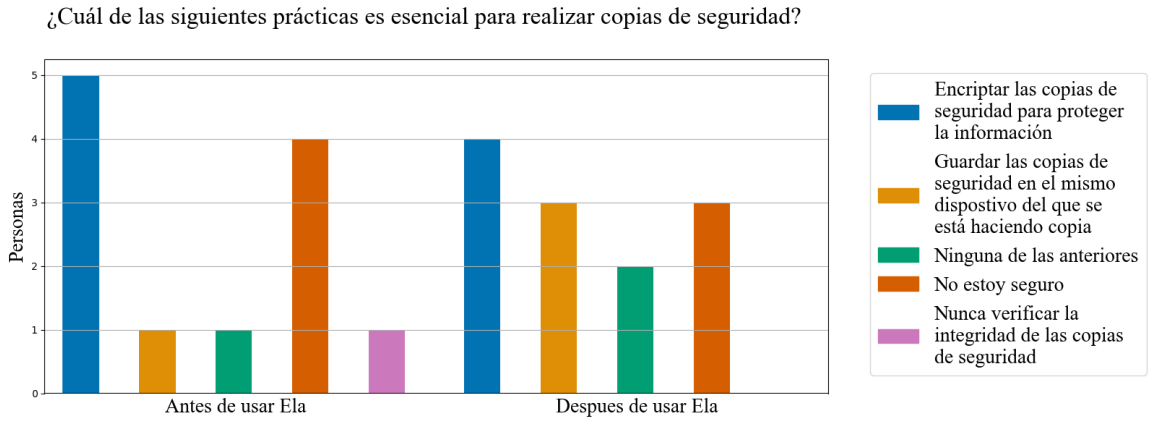


Figura 30: Evolución de las pregunta de conocimientos de ciberseguridad sobre copias de seguridad antes y después de haber usado Ela

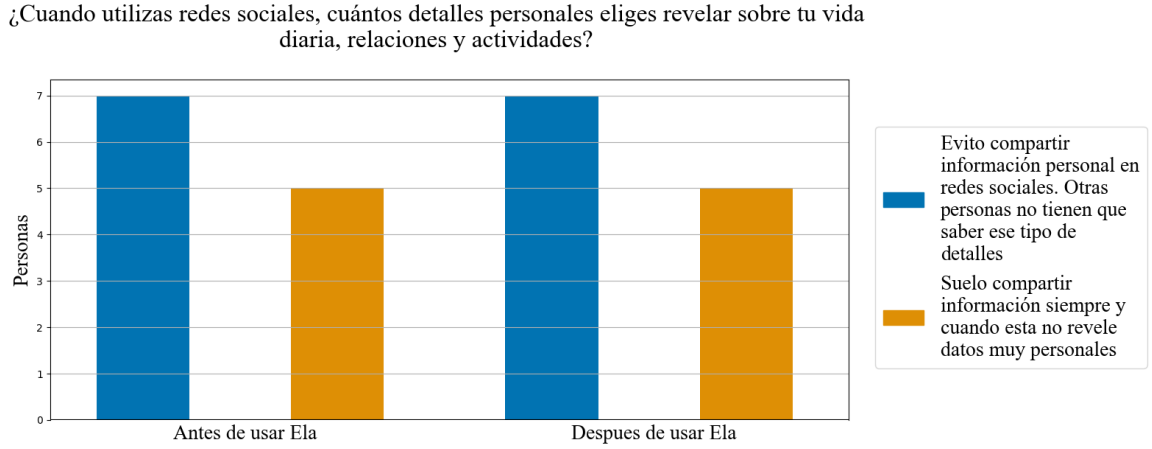


Figura 31: Evolución de las pregunta de conocimientos de ciberseguridad sobre uso de redes sociales antes y después de haber usado Ela

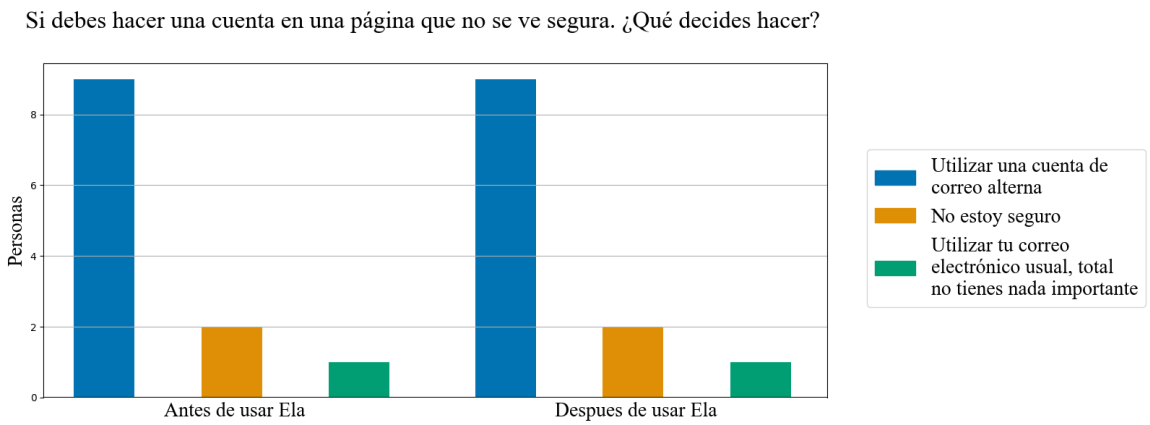


Figura 32: Evolución de las pregunta de conocimientos de ciberseguridad sobre páginas inseguras antes y después de haber usado Ela

Finalmente, como se puede notar en las figuras 31 y 32, el uso de Ela no afectó en la manera que los participantes interactuaban con sus redes sociales ni en otro tipo de páginas *web*.

CAPÍTULO 8

Conclusiones

La aplicación diseñada fue efectiva en evitar que el usuario accediera a dominios maliciosos. Esto era más efectivo cuando después de bloquear el acceso a estos dominios, se mostraba un mensaje sobre los peligros de ese dominio y cómo podría defenderse contra esa amenaza. Esto aumentaba en 50 % la probabilidad de que se defendieran contra este tipo de ataques.

Por otro lado, la aplicación fue capaz de educar sobre ciberseguridad. Sin embargo, no tuvo la misma efectividad en todas las áreas. Era particularmente efectiva para educar sobre los diferentes grupos de *malware* que existían y otros aspectos como autenticación multifactor. Pero no era tan efectiva para transmitir información como copias de seguridad e ingeniería social.

Posteriormente, la aplicación logró identificar qué aplicaciones maliciosas un usuario tenía instalado en su teléfono, pero esto más que causarles preocupación; les causaba interés sobre qué tan peligrosos eran los permisos que se le brindaban a la aplicación para que fuera considerada sospechosa.

Finalmente, los usuarios interactuaron principalmente con el módulo del asistente virtual. Aunque con el paso del tiempo las personas participantes dejaron de interactuar tanto con el asistente, introducir el módulo del dato curioso de ciberseguridad causó que el 33 % de los usuarios siguieran interactuando frecuentemente con el asistente.

CAPÍTULO 9

Recomendaciones

- Extender la protección de dominios maliciosos determinando cuál aplicación intentó hacer la conexión. Luego, apoyarse de estos resultados para la identificación de aplicaciones sospechosas.
- Al realizar los ataques de *phishing* simulados, tomar en cuenta que Google cuenta con varios sistemas para detectar *phishing*. Es bastante probable que los dominios de *phishing* sean bloqueados directamente por el navegador en menos de 24 horas.
- Además de utilizar inteligencia artificial para detectar dominios maliciosos, también apoyarse de otras fuentes confiables como Google Safe Browsing.
- Diversificar la cantidad de plataformas en las que puede correr en proyecto. Específicamente, se recomienda centrarse en otros dispositivos móviles como iOS.
- Para filtrar el tráfico de red se recomienda en enfocarse en protocolos no cifrados. Si aún así se desea observar tráfico cifrado, se puede instalar manualmente una autoridad certificadora. Sin embargo, instalar una autoridad es un procesoe que el usuario debe realizar manualmente; no se puede automatizar a través de código en las versiones de Android actuales.
- Por razones morales y legales, es de vital importancia asegurarse que los usuarios estén de acuerdo de participar en una prueba de ataque simulado antes de ejecutarlas.

CAPÍTULO 10

Bibliografía

[1] S. J. Adam McNeil, *Mobile Malware Threats Are Surging in Europe*, en-us. dirección: https://www.proofpoint.com/us/blog/email-and-cloud-threats/mobile-malware-surging-europe-look-biggest-threats.

[2] P. B. G. y. G. G. Abraham Silberschatz, *Operating System Concepts Essentials*. Wiley, 2014.

[3] R. Garg y G. Verma, *Operating Systems : An Introduction*. Mercury Learning & Information, 2017, ISBN: 9781942270386. dirección: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso&db=e000xww&AN=1809145&lang=es&site=ehost-live&scope=site&custid=s4224794.

[4] Y. Mehdi, "Windows 10: Powering the world with 1 billion monthly active devices," 2020. dirección: https://blogs.windows.com/windowsexperience/2020/03/16/windows-10-powering-the-world-with-1-billion-monthly-active-devices/.

[5] E. Protalinski, *Chromebooks outsold Macs worldwide in 2020, cutting into Windows market share*, en-us. dirección: https://www.geekwire.com/2021/chromebooks-outsold-macs-worldwide-2020-cutting-windows-market-share/.

[6] Microsoft, "Meet Windows 11," dirección: https://www.microsoft.com/en-us/windows/windows-11.

[7] B. Wolfe, *Apple macOS versions: everything you need to know*, en-us. dirección: https://www.techradar.com/news/apple-macos-versions-everything-you-need-to-know.

[8] Android, "Android 14," 2023. dirección: https://www.android.com/android-14.

[9] A. Developers, *Permissions on Android*, en-us. dirección: https://developer.android.com/guide/topics/permissions/overview.

[10] A. W. Tanenbaum, *Structured Computer Organization*. Pearson Education, 2005, ISBN: 0-13-148521-0.

[11] K. Ewusi-Mensah, *Software Development Failures*. The MIT Press, 2003, ISBN: 0262050722. dirección: https://books.google.com.gt/books?id=cWde_yxJorEC&lpg=PR9&ots=WeWw-xoON2&dq=software%20development&lr&pg=PR4#v=onepage&q&f=false.

[12] G. E. N. Andrew Butterfield y A. Kerr, *A Dicrionary of Computer Science*. Oxford University Press, 2016, ISBN: 9780199688975.

[13] M. John C., *Concepts in Programming Languages*. Cambridge University Press, 2003, ISBN: 9780521780988. dirección: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso&db=e000xww&AN=112510&lang=es&site=ehost-live&scope=site&custid=s4224794.

[14] R. Hat, *¿Qué es una API y cómo funciona?* Dirección: https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces.

[15] J. R. Capacho Portilla y W. Nieto Bernal, *Diseño de base de datos*. Universidad del Norte, 2017, ISBN: 9789587418255. dirección: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso&db=e000xww&AN=1690049&lang=es&site=ehost-live&scope=site&custid=s4224794.

[16] P. Pacheco, *An Introduction to Parallel Programming*. Morgan Kaufmann, 2011, ISBN: 978-0-12-374260-5.

[17] IBM, *The Fundamentals of Networking*, en-us. dirección: https://www.ibm.com/topics/networking.

[18] S. S. Shinde, *Computer Network*. New Age International, 2009, ISBN: 9788122425772. dirección: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso&db=e000xww&AN=316072&lang=es&site=ehost-live&scope=site&custid=s4224794.

[19] Cisco, *What is IPv6?* en-us. dirección: https://www.cisco.com/c/en/us/solutions/ipv6/overview.html.

[20] V. Vij, *Computer Networks*. Laxmi Publications Pvt Ltd, 2018, ISBN: 9789352740802. dirección: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso&db=e000xww&AN=2228701&lang=es&site=ehost-live&scope=site&custid=s4224794.

[21] Cloudflare, *IPv6 Adoption*, en-us, 2022. dirección: https://radar.cloudflare.com/reports/ipv6.

[22] Cloudflare, *What is a subnet*, en-us. dirección: https://www.cloudflare.com/learning/network-layer/what-is-a-subnet/.

[23] K. W. R. James F. Kurose, *Computer Networking: A Top-Down Approach, 7th Edition*. Pearson, 2017, ISBN: 0133594149.

[24] L. L. Peterson y B. S. Davie, *Computer Networks : A Systems Approach*. (The Morgan Kaufmann Series in Networking). Morgan Kaufmann, 2007, vol. 4th ed, ISBN: 9780123705488. dirección: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso&db=e000xww&AN=320736&lang=es&site=ehost-live&scope=site&custid=s4224794.

[25] R. Moskowitz, D. Karrenberg, Y. Rekhter, E. Lear y G. J. de Groot, *Address Allocation for Private Internets*, RFC 1918, feb. de 1996. DOI: 10.17487/RFC1918. dirección: https://www.rfc-editor.org/info/rfc1918.

[26] Cloudflare, *What is a domain name?* en-us. dirección: https://www.cloudflare.com/learning/dns/glossary/what-is-a-domain-name/.

[27] IETF, en. dirección: https://www.ietf.org/standards/rfcs/.

[28] Cloudfalre, *What is UDP?* en-us. dirección: https://www.cloudflare.com/learning/ddos/glossary/user-datagram-protocol-udp/.

[29] M. W. Docs, *An overview of HTTP*, en-us. dirección: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview.

[30] C. Weider, *Resource Transponders*, RFC 1728, dic. de 1994. DOI: 10.17487/RFC1728. dirección: https://www.rfc-editor.org/info/rfc1728.

[31] L. Daigle, *WHOIS Protocol Specification*, RFC 3912, sep. de 2004. DOI: 10.17487/RFC3912. dirección: https://www.rfc-editor.org/info/rfc3912.

[32] Cisco, *What is a Virtual Private Network (VPN)?* en-us. dirección: https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html.

[33] G. C. P. y G. K. K., *Cybersecurity : A Self-Teaching Introduction.* Mercury Learning & Information, 2020, ISBN: 9781683924982. dirección: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso&db=e000xww&AN=2399513&lang=es&site=ehost-live&scope=site&custid=s4224794.

[34] M. Security, *What is malware?* en-us. dirección: https://www.microsoft.com/en-us/security/business/security-101/what-is-malware.

[35] A. S. Research, *Flagging 13 Million Malicious Domains in 1 Month with Newly Observed Domains*, en-us. dirección: https://www.akamai.com/blog/security-research/newly-observed-domains-discovered-13-million-malicious-domains.

[36] B. Gyunka y O. Abikoye, "Analysis of Human Factors in Cyber Security: A Case Study of Anonymous Attack on Hbgary," *Computing and Information Systems Journal*, vol. 21, págs. 10-18, ene. de 2017.

[37] L. González-Manzano y J. M. de Fuentes, "Design recommendations for online cybersecurity courses," *Computers & Security*, vol. 80, págs. 238-256, 2019, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2018.09.009. dirección: https://www.sciencedirect.com/science/article/pii/S0167404818302050.

[38] Google, *Helping kids be safe, confident explorers of the online world.* en-us. dirección: https://beinternetawesome.withgoogle.com/en_us.

[39] "Google Keynote (Google I/O'19)," 2019.

[40] P. Bhat y K. Dutta, "A survey on various threats and current state of security in Android platform," *ACM Computing Surveys*, vol. 52, n.º 1, págs. 1-35, feb. de 2019. DOI: 10.1145/3301285. dirección: https://doi.org/10.1145/3301285.

[41] P. N. Suart Russel, *Artificial Intelligence: A Modern Approach.* Pearson Education, 2010, ISBN: 0136042597.

[42] IBM, *What is an AI model?* en-us. dirección: https://www.ibm.com/topics/ai-model.

[43] C. Drumond, *What is Scrum And How to get Started?* en-us. dirección: https://www.atlassian.com/agile/scrum.

[44] D. Radigan, *Kanban*, en-us. dirección: https://www.atlassian.com/agile/kanban.

[45] V. Soinien, *Jetpack Compose vs React Native - Differences in UI development*, en-us. dirección: https://www.theseus.fi/bitstream/handle/10024/507066/Soininen_Visa.pdf?sequence=2.

[46] D. Z. Corado, *Desarrollo de Asistente Virtual para el acompañamiento en la navegación de sitios web o descarga de aplicaciones de forma segura*, 2023.

[47] en, sep. de 2023. dirección: `https://www.merriam-webster.com/dictionary/alphanumeric`.

CAPÍTULO 11

Anexos

11.1. Capturas de pantallas de la aplicación

Figura 33: Pantalla de inicio al tener varias aplicaciones sospechosas

Figura 34: Pantalla de inicio al haber bloqueado un dominio en ese día

Figura 35: Pantalla de ajustes

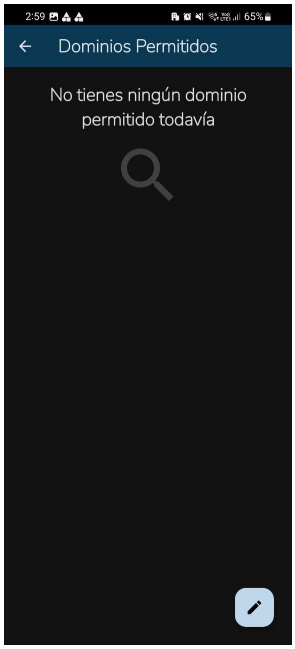


Figura 36: Pantalla de agregar un dominio permitido vacía

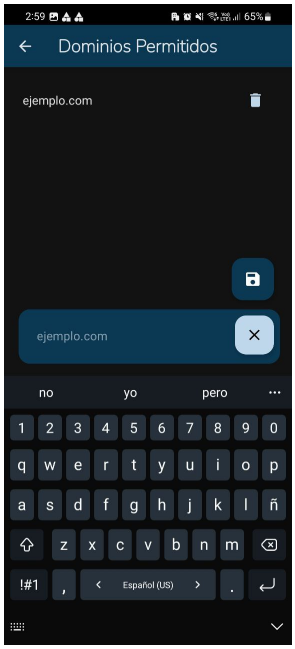


Figura 37: Pantalla de agregar un dominio permitido

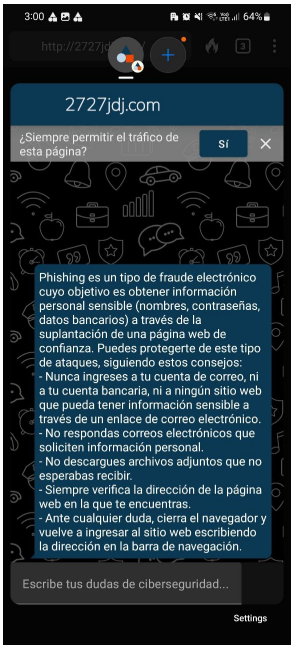


Figura 38: Notificación al bloquear un dominio (versión burbuja)

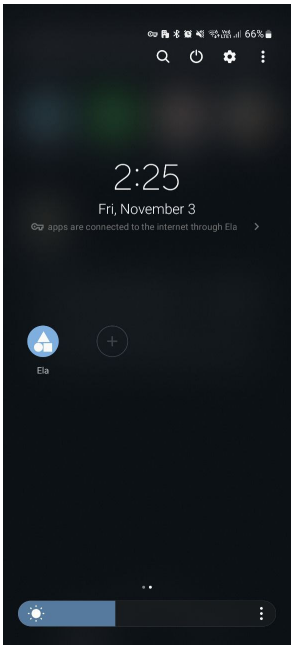


Figura 39: Botón de acceso rápido para encender a Ela

11.2. Encuestas realizadas

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

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 considera que es su nivel de conocimiento en ciberseguridad? *

Marca solo un óvalo.

	1	2	3	4	5	
No s						Soy un experto

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

Marca solo un óvalo.

	1	2	3	4	5	
No e	◯	◯	◯	◯	◯	Es muy importante

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

Marca solo un óvalo.

	1	2	3	4	5	
No i	◯	◯	◯	◯	◯	Influye mucho

Conocimientos

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
- ◯ Otros: _____

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

Selecciona todas las opciones que correspondan.

- [ ] Pepito123$, es segura porque tiene más de 8 caracteres, minúsculas, mayúsculas, números y símbolos
- [ ] c0ntr4seni4s, es segura porque utiliza una combinación de números y letras
- [ ] JesusTeAmo, 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 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:

Marca solo un óvalo.

- ◯ Spam
- ◯ Troyano
- ◯ Doxing
- ◯ Ransomware
- ◯ No estoy seguro

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.

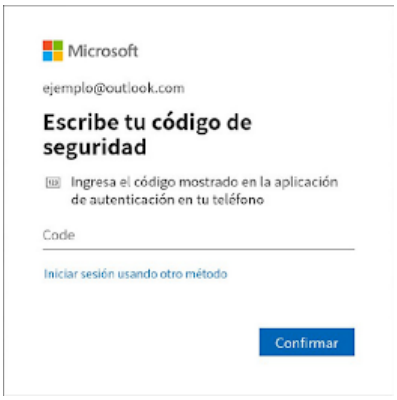
- Me encanta compartir mi día a mis amigos en las redes sociales, por algo es una red social
- Suelo compartir información siempre y cuando esta no revele datos muy personales
- Evito compartir información personal en redes sociales. Otras personas no tienen que saber ese tipo de detalles
- No uso redes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

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

Marca solo un óvalo.

Opción 1

Opción 2



Opción 3

Opción 4

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 dispostivo del que se está haciendo copia
- Ninguna de las anteriores
- No estoy seguro

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

Noviembre, 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 15 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

CAPÍTULO 12

Glosario

alfanumérico: Formado por letras y números y un grupo de símbolos reducidos como guión y guión bajo.[47]. 19

bit: Es una unidad que únicamente cuenta con dos estados; usualmente 0 y 1. Es parte fundamental del sistema numérico binario. 20

código abierto: Es un modelo de desarrollo de programas, donde los desarrolladores comparten públicamente el código utilizado para crear el programa. 11

ela: El nombre de la aplicación desarrollada en este trabajo. 32, 37, 39, 46, 48, 51, 52, 55, 56

encabezado: Parte de un paquete que contiene metadatos de la información en tránsito. 17

interfaz: Una conexión física o lógica en con la que un dispositivo se conecta a una red. 21

metadatos: Son datos que describen a otros datos. 21

nodo: En redes, se refiere a cualquier dispositivo que puede recibir y enviar información. 22

octeto: Es un grupo de ocho bits o un byte. Su representación numérica en binario puede ir de 00000000 (0 en sistema decimal) hasta 11111111 (255 en sistema decimal). 17, 18