UNIVERSIDAD JOSÉ ANTONIO PÁEZ

DESARROLLO DE UNA APLICACIÓN PARA DETECTAR Y ELIMINAR VIRUS DE ORDENADORES

Autores:
Castelli Marialejandra
Pérez Diego

Urb. Yuma II, calle Nº 3. Municipio San Diego
Teléfono: (0241) 8714240 (master) – Fax: (0241) 8712394

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN
CARRERA INGENIERÍA DE COMPUTACIÓN

DESARROLLO DE UNA APLICACIÓN PARA DETECTAR Y ELIMINAR VIRUS DE ORDENADORES

Proyecto del Trabajo de Grado para optar al título de

INGENIERO DE COMPUTACIÓN

Autores:
Diego Pérez
C.I.26.369.224
Marialejandra Castelli
C.I.26.581.841
Tutor: Ing. Belkys Araujo

San Diego, Junio de 2020

FI-C-011-2020-1CR (TG)

Valencia, 19 de junio de 2020

Ciudadanos:
Castelli R., Marialejandra.
26.581.841
Perez G., Diego A.
26.369.224
Presente-

Cumplo con informarle que la Comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión **Nº 04-2020** de fecha **13-02-2020** aprobó el proyecto de trabajo de grado titulado ***DESARROLLO DE UNA APLICACIÓN PARA DETECTAR Y ELIMINAR VIRUS DE ORDENADORES*** presentado por usted (es) como requisito para optar al título de Ingeniero en Computación.

Se ratifica la designación de la Ing. Belkys Araujo C.I: 6.906.234 como Tutora Académica que los asesorara en el desarrollo de este proyecto.

Atentamente,

Prof. Luís Lira

Decano de la Facultad de Ingeniería

c.c. Coordinación de Pasantías y Trabajo de Grado (1).

Ll/a.a.

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERIA
ESCUELA DE COMPUTACIÓN

APROBACIÓN DEL TUTOR

Quien suscribe, Ingeniero Belkys Araujo, portadora de la cédula de identidad **Nº**6.906.234, en mi carácter de tutor del trabajo de grado presentado por los ciudadanos: **Marialejandra Castelli,** portadora de la cédula de identidad **Nº**26.581.841**,** y **Diego Pérez**, portador de la cédula de identidad **N°**26.369.224 , titulado**, DESARROLLO DE UNA APLICACIÓN PARA DETECTAR Y ELIMINAR VIRUS DE ORDENADORES.,** presentado como requisito parcial para optar al título de **INGENIERO EN COMPUTACIÓN**, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe

En San Diego, a los 3 días del mes de julio del año dos mil veinte.

_____

Ing. Belkys Araujo

C.I.: 6.906.234

AGRADECIMIENTOS

Este presente trabajo de grado es gracias a las siguientes personas:

A mis abuelos maternos, gracias por apoyarme durante todo mis trascurso durante estos últimos 10 semestres.

A mi padre y abuela paterna, a mi abuela gracias por creer en mí y motivarme siempre, tanto dentro como fuera de la universidad, a mi padre por apoyarme a su manera en todo este trascurso.

A mi madre, por todas esas idas a la universidad y demás sacrificios que hizo durante todo mi trascurso.

A mis tíos, por ser como unos segundos padres para mí y ser mi inspiración para estudiar esta carrea.

Y finalmente a mi abuelo Antonio a pesar de que no está aquí a él le hubiera gustado verme aquí y sé que me hubiera apoyado al 100 %.

A todos, muchas gracias.

Diego Pérez.

AGRADECIMIENTOS

Principalmente, estoy muy agradecida con Dios, toda mi familia y mis amigos que me apoyaron durante mi carrera y este trabajo de grado:

A mi madre, por siempre estar a mi lado, por apoyarme y aconsejarme en mi travesía universitaria y hacer su máximo esfuerzo para que pudiera estudiar esta carrera.

A mi padre, por hacer lo posible por ayudarme en mi transcurso y velar por mi seguridad.

A mi abuela y mi abuelo, por siempre estar atentos y preocuparse por mí. Por toda la ayuda que me han brindado en todo este tiempo, no solo en la carrera, sino también en mi vida.

A mi hermana, por darme apoyo y ánimo durante este recorrido.

A mi tío, que a pesar de que desde hace mucho tiempo que no se encuentra físicamente conmigo, se que siempre ha estado a mi lado cuidándome.

A la profesora Belkys Araujo, quien nos asesoró y guió como nuestra tutora en este proyecto tan especial.

A todos mis amigos, a los que conocía desde antes de entrar a la universidad y a los que conocí en la institución, gracias por apoyarme y compartir esta etapa conmigo.

Muchas gracias.

Marialejandra Castelli

ÍNDICE

LISTA DE TABLAS

TABLA **Pág.**

LISTA DE FIGURAS

TABLA **Pág.**

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN
CARRERA INGENIERÍA DE COMPUTACIÓN

DESARROLLO DE UNA APLICACIÓN PARA DETECTAR Y ELIMINAR VIRUS DE ORDENADORES

Autores: Diego Pérez y Marialejandra Castelli
Tutor: Ing Belkys Araujo
Fecha: Mayo de 2020

RESUMEN

El presente trabajo propone el desarrollo de una aplicación para detectar y eliminar virus de ordenadores, mediante el uso de la ingeniería de software y Lenguaje de Programación (Python). La investigación comprende un proyecto especial, con diseño descriptivo y nivel de la investigación de campo, así mismo la población corresponde a la mayoría de los antivirus disponibles en el mercado siendo utilizado más adelante una muestra de (6) antivirus para obtener mediante la observación directa y una lista de cotejo las características y requerimientos de los mismos. Para el desarrollo de la presente aplicación, se propuso el uso de la metodología de desarrollo de software prototipo. El presente trabajo consta de 5 fases metodológicas las cuales son: el diagnostico en el cual se hará una comparación de los diferentes antivirus del mercado, la fase de determinación en esta fase en función de lo obtenido en la fase anterior se determinara cuáles son las características y requerimientos básicos de un antivirus, la fase diseño y desarrollo en las cuales de diseñar y desarrollara la aplicación mediante el modelo de prototipos y el lenguaje de programación Python respectivamente, la última fase es el periodo de prueba en la cual se evaluara en funcionamiento de la aplicación mediante las pruebas de caja negra y caja blanca. Con la realización de este trabajo de grado, se concluye que se puede realizar un antivirus utilizando el leguaje Python, haciendo uso de los hash y códigos MD5 y la librería gráfica Tkinter, capaz de detectar virus, gusanos y troyanos, y eliminarlos.

Descriptores: Antivirus, virus, MD5

INTRODUCCIÓN

En los últimos años, la dependencia de un usuario ya sea persona o una empresa sobre un ordenador a incrementado enormemente, esto debido a la cantidad de tareas que están cada vez más ligados a los mismos y que se encuentran en aumento a la vez que se les son facilitadas a los usuarios. Un ejemplo de esto serían las operaciones bancarias que al poder hacerla a distancia le facilitan la tarea al usuario. Otra de las razones de porque los ordenadores se han vuelto tan necesarios, es la cantidad de información que están ligados a los mismos (cuentas bancarias, claves, documentos importantes, entre otros). Además, cabe resaltar que un ordenador puede realizar tareas que una persona no, es por esto que la gran mayoría de las empresas usan ordenadores en el día a día. La gran dependencia ha hecho a los ordenadores extremadamente importantes a tal punto que la pérdida de uno de estos o de su información almacenada causaría un gran daño y no solo en el ámbito financiero.

Si bien los ordenadores son máquinas sorprendentes por todo lo que pueden lograr y lo importantes que son, no son perfectos ya que estos son vulnerables a ataques de virus informáticos. Al principio estos virus tenían un efecto mínimo, pero con el pasar de los años sus efectos y consecuencias fueron escalando hasta el punto de acabar por completo con un ordenador. Por estas razones, se crearon programas capaces de detectar y eliminar estos virus, cuya finalidad es proteger el ordenador del usuario. Estos programas varían entre que tan eficientes son en su cometido, por lo general los menos eficientes son aquellos que se pueden conseguir de manera gratuita y los más eficientes son los pagos, es por esto que la mayoría de los usuarios que no tienen como costear uno de calidad optan por los menos eficientes.

Por este motivo, en el presente trabajo se plantea el desarrollo de un software de antivirus mediante el uso de la inteligencia artificial, que funcione como una alternativa para ser utilizado en el lugar de otros programas gratuitos en el mercado. Con ese objetivo como meta, se realizó y estructuró este trabajo de grado según se indica a continuación:

En el Capítulo I, se plantea la problemática y las interrogantes de la investigación, de los cuales deriva el objetivo general, los objetivos específicos, la justificación y el alcance de la investigación.

En el Capítulo II, se presentan los antecedentes que sustentaron la investigación.

En el Capítulo III, que contiene la metodología que se utilizó, el tipo y diseño de la investigación, la población y muestra, las técnicas e instrumentos de recolección de datos, y las fases metodológicas.

En el Capítulo IV, se describe el análisis y los resultados durante la realización del trabajo de grado.

En el Capítulo V, se presentan las conclusiones y recomendaciones de la investigación.

CAPÍTULO I
EL PROBLEMA

1.1.Planteamiento del problema

El término malware proviene de la contracción del inglés malicious software, que en español se traduce a software malicioso, y se utiliza para todo aquel programa o código que signifique un riesgo para la computadora, más específicamente, cuya finalidad sea provocar un mal funcionamiento, malograr sistemas o datos almacenados y hasta robar información. Este término fue utilizado por primera vez en la década de los 90, con la intención de hacer una mejor cobertura al término de virus informático. En los tiempos actuales, es prácticamente imposible que no sea considerado un riesgo al momento de utilizar alguna tecnología.

Dicho software puede presentarse de diferentes formas, las cuales han ido aparecido y evolucionado a lo largo del tiempo, haciendo que su propagación sea cada vez más rápida y, en ciertos casos, mucho más dañina. Los malware más comunes con los que el usuario ordinario puede encontrarse son los virus, el caballo de Troya, el spyware o programa espía y los gusanos.

En el caso de los virus, uno de los primeros malware conocidos y cuya principal característica es la capacidad de replicarse a sí mismo dependiendo siempre de un archivo, en un principio solo se abstenían a mostrar mensajes en pantalla o consumir cierta cantidad de los recursos del sistema pero, a medida que los años fueron pasando y las tecnologías se transformaron y se volvieron cada vez más poderosas, este tipo de malware se adaptó de tal forma que logró alarmar tanto al consumidor común como a las grandes empresas por lo dañino que podría llegar a ser. Si bien, hoy en día esta clase de virus ha disminuido en su proliferación para darle el paso a otros malware más complejos y, por consecuencia, más peligrosos.

Los worms o gusanos, son casi tan antiguos como los virus y muchas veces son llamados de igual forma que estos. Su característica fundamental que los diferencia de

un virus, es que no tiene dependencia de otro archivo para su ejecución. Su propagación es otro de sus aspectos problemáticos, ya que pueden esparcirse por la red, por correo electrónico, mensajería instantánea o dispositivos USB. Usualmente, se incorporan al sistema de inicio del computador para ser cargados, y de esta forma, colapsan los ordenadores y redes de las víctimas.

Este tipo de malware, la mayoría de las veces, se utiliza para transportar otro tipo de software malicioso tomando como ventaja las vulnerabilidades del dispositivo al que está atacando. A pesar de esto, los worms, por la misma razón de que no están adheridos a otro archivo, su eliminación del sistema es más fácil que la de otros tipos de malware.

Por otro lado, los software maliciosos encargados de recolectar información de los usuarios son llamados Spyware o programas espías. Su finalidad es obtener datos de las personas que utilicen el dispositivo, del sistema o hasta de dispositivos de almacenamiento sin que las víctimas tengan consciencia de ello. A diferencia de los otros malware ya nombrados, estos no se auto propagan, sino que solo pueden entrar al sistema al ser instalados a través de Internet. Además, raras veces le generan daños al sistema.

Cabe destacar que, en algunas ocasiones, las consecuencias que pueda ocasionar un malware a un equipo o a una red no son percibidas por el usuario inmediatamente. Muchos troyanos, los cuales se encargan de aparentar normalidad en el computador mientras se alojan en algún archivo, y virus se introducen en los sistemas para recoger información de sus víctimas, enviar correos electrónicos masivos o hasta para realizar conexiones remotas sin llamar la atención del consumidor. Esto permite que el delincuente continúe sus actividades ilícitas mientras que la víctima no tome medidas para contrarrestarlas.

Tomando en cuenta toda la información que una persona puede tener en la computadora y lo indispensable que esta pueda ser en la vida del usuario, el que esta se dañe, se elimine, se destruya o sea robada podría suponer una gran pérdida de dinero o de algún recurso importante, como un archivo o una base de datos. En algunos casos,

el ataque de un malware hasta puede significar la completa inutilización del equipo, lo cual aumentaría el gasto para la reparación o compra de uno nuevo.

La forma más efectiva de contrarrestar estos software maliciosos es haciendo uso de un antivirus. Pero, tomando en cuenta la situación económica en la que se encuentra el país, una gran cantidad de usuarios comunes no puede costear un antivirus como medida para la eliminación y prevención contra los malware. Esto es debido a que se encuentran con el obstáculo de que la mayoría solicita un pago en dólares o, en su defecto, las empresas exigen el cambio en bolívares soberanos, que no siempre es un precio fiable ya que la compañía distribuidora puede cambiar el costo según se deprecie el bolívar frente al dólar.

Por otro lado, al no tener capital para comprar un antivirus, los usuarios prefieren optar por un servicio gratuito que no siempre brinda la protección adecuada y necesaria para el equipo. Como resultado, los consumidores se vuelven susceptibles a caer víctima de piratas informáticos y, como consecuencia, corren el riesgo de que su seguridad se vea comprometida. Es debido a esto, que se propone la realización de un software de antivirus gratuito que sea capaz de identificar una amenaza, analizarla y eliminarla, sirviendo como una forma de protección dirigida al mercado de bajos recursos.

1.2. Formulación del problema

¿Cómo se puede desarrollar un antivirus que identifique y elimine software malicioso y que funcione como una alternativa a los demás modelos gratis disponibles?

1.3. Objetivos de la investigación

1.3.1 Objetivo general

Desarrollar un antivirus mediante ingeniería del software que funcione como alternativa a sus semejantes gratuitos.

1.3.2 Objetivos específicos

1. Realizar un diagnóstico comparativo entre las características necesarias según los entrevistados mediante una lista de cotejo.

2. Determinar los requerimientos funcionales y no funcionales en función al diagnóstico anteriormente realizado.

3. Diseñar el antivirus utilizando el modelo de prototipos.

4. Desarrollar el antivirus mediante la utilización del lenguaje Python.

5. Probar el funcionamiento del antivirus mediante las pruebas de caja negra y caja blanca.

1.4 Justificación de la investigación

Los antivirus tienen como principales objetivos el evitar, identificar y eliminar los diferentes tipos de malware que puedan abrirse el paso dentro de las computadoras de los usuarios. Con la proliferación de nuevos y más poderosas clases de software malicioso, emana la inevitable búsqueda por una forma de protección adecuada para prevenir comprometer información privada o crucial, aludiendo tanto a usuarios individuales como empresas.

El poseer una medida que permita que los ordenadores estén protegidos contra los ataques de programas maliciosos traerá como beneficio a los usuarios estar protegidos contra robo de información o identidad, ataques de terceros que dañen permanentemente al ordenador, pérdida masiva de información, entre otros. Estas ventajas son aplicables tanto en nivel local, como lo sería la ciudad de valencia, como en el resto del país debido al gran uso que tiene un ordenador en la vida diaria de personas y empresas. Es por este mismo uso que se está más expuesto a los programas maliciosos y todas las consecuencia que estos traen, por consiguiente los beneficios que traerá el presente programas son aún mayores.

El desarrollo e implementación del presente proyecto permitirá a todos los ciudadanos y empresas de Venezuela tener una alternativa viable y efectiva para proteger sus ordenadores. Por otra parte, dicho proyecto traerá prestigio a la universidad José Antonio Páez no solo por la complejidad que conlleva el desarrollo de un antivirus, sino que también el desarrollo de seguridad en ordenadores es una de las tecnologías en la cual se está haciendo énfasis en la actualidad. De esta forma, el

desarrollo exitoso del software presentado servirá de incentivo para futuros estudiantes el tomar como ideas de Trabajo de Grado temas innovadores.

Por otro lado, los aspectos favorables de este programa no solo se centran en seguridad, sino que también se evidencia en el aspecto económico, ya que actualmente adquirir un antivirus de calidad tiene un costo bastante elevado que muchos usuarios del país no pueden costear, gracias a que el software de antivirus propuesto será gratuito, se le está brindado a todos esos usuarios la capacidad de contar con protección eficiente para sus ordenadores.

1.5. Alcance y limitaciones

El software antivirus usa dos métodos para proteger el sistema. El primero es analizar los archivos almacenados en la computadora, comparándolos con una base de datos de software o programas malignos. El segundo es la monitorización constante del comportamiento de los archivos del sistema que puedan estar infectados.

Con respecto al análisis de archivos, el antivirus compara cada archivo del disco duro con una base de datos que tiene registro de los virus o malware ya conocidos. Si cualquier pieza de código en un archivo del disco duro coincide con el virus conocido en los registros, el software antivirus entra en acción, llevando a cabo una de las siguientes acciones posibles:

- Reparar el archivo: donde el antivirus trata de reparar el archivo infectado eliminando el virus.
- Ponerlo en cuarentena: el antivirus intentará proporcionar protección contra el virus, haciendo inaccesibles los programas a este archivo, impidiendo su propagación y ejecución.

CAPÍTULO II
MARCO TEÓRICO

El estudio debe de ser apoyado por un marco teórico, el cual ha sido definido por Arias (2012) como “el producto de la revisión documental-bibliográfica, y consiste en una recopilación de ideas, posturas de autores, conceptos y definiciones, que sirven de base a la investigación a realizar”. (p.106). Siguiendo esta idea, Supo (2015) agrega que “debe contener cada palabra o frase que aparece (…) en el enunciado (…), debidamente ordenados y estructurados”. (p.27.).

Tomando lo anteriormente manifestado como un punto importante, en el presente trabajo se utilizó este capítulo para especificar las bases teóricas sobre las que se sostiene el estudio.

2.1. Antecedentes

Los antecedentes conforman una base fundamental para todo estudio, ya que expresan los resultados de proyectos referentes al tema y por tanto “reflejan los avances y el estado actual del conocimiento en un área determinada y sirven como modelo o ejemplo para futuras investigaciones” (Arias, 2012, p.106). A continuación, se presentan los trabajos que sirvieron como antecedente por su relación con el tema trabajado en este trabajo de investigación.

En primer lugar, Ruano-Ordás (2016), cuyo artículo para la revista Iberoamericana de Inteligencia Artificial tiene como titulo **Resumen de Tesis: Modelo para la optimización de la ejecución de filtros anti-spam**, expresó en su trabajo que era necesario el reducir el excesivo uso de los recursos computacionales, incrementar la velocidad de filtrado y ajustar los pesos empleados para la combinación de diversas técnicas de filtrado en correos. Dicha investigación se relaciona indirectamente con el trabajo planteado con su aporte con respecto a la implementación de las técnicas de filtrado. Estas técnicas sirven como referencia para el desarrollo de la programación del filtrado de archivos potencialmente peligrosos que debe de utilizar el software de antivirus que se dispuso realizar en este trabajo.

Por otro lado, Rodríguez, Oduber y Mora (2017), en el artículo **Actividades rutinarias y cibervictimización en Venezuela** que realizaron para la Revista Latinoamericana de Estudios de Seguridad, expresan que la victimización en línea ha sido estudiada a fondo por países desarrollados para recopilar información y evaluarla, con la finalidad de predecir y explicar cómo se efectúan los delitos relacionados con la tecnologías y comunicaciones.

Con el enfoque a la realidad venezolana dado por el artículo anterior, se obtiene una idea clara de las rutinas diarias en las que el usuario se encuentra más vulnerable a sufrir un ataque informático, permitiendo así que a la hora de la realización del antivirus se tenga una noción precisa de en cuales filtros del programa se deberá de enfocar el software.

Así mismo, Islam (2018), cuyo trabajo de grado para optar por su título de Master de Ciencias en Tecnología tiene como título traducido **Una solución orquestada de prueba de productos anti-malware para entornos conectables múltiples**, propuso un concepto de solución orquestada que satisficiera las necesidades del equipo de investigación de seguridad con respecto a la automatización del proceso de prueba y análisis del anti-malware. Si bien, la solución no pudo ser implementada, es muy importante tomar en cuenta la idea que plantea el ya mencionado trabajo, ya que busca la simplificación y rapidez dentro del ámbito.

Por último, Laura y Tumi (2019), en el trabajo de investigación titulado **Sistema antivirus multiplataforma en tiempo real usando técnicas heurísticas y proactivas** adaptado para la revista Ciencia, Tecnología y Desarrollo de la Universidad de Altiplano, propusieron que la seguridad informática como investigación tiene como principal labor la seguridad de la información y estabilidad del sistema operativo host, evitar que programas sospechosos y/o potencialmente peligrosos se infiltren en el sistema.

El artículo presentado colabora de manera directa con el presente trabajo de grado debido a que las técnicas heurísticas y proactivas utilizadas en la investigación permitirán simplificar la tarea vital de programar al antivirus en desarrollo, de forma

que identifique cuales programas debe detectar y, de la misma manera, cual o cuales de los mencionados programas debe eliminar.

2.2. Bases Teóricas

Bavaresco (2013) sostiene que, “las bases teóricas tienen que ver con las teorías que brindan al investigador el apoyo inicial dentro del conocimiento del objeto de estudio, es decir, cada problema posee algún referente teórico, lo que indica, que el investigador no puede hacer abstracción por el desconocimiento, salvo que sus estudios se soporten en investigaciones puras o bien exploratorias”. (p.51). Por otra parte, Arias (2012) afirma que “Las bases teóricas implican un desarrollo amplio de los conceptos y proposiciones que conforman el punto de vista o enfoque adoptado, para sustentar o explicar el problema planteado”. (p. 107). En función de estas definiciones se entiende que las bases teóricas es todo el desarrollo extendido de todos los conceptos que sustentaran y explicaran el problema planteado.

La presente investigación se enfoca en el Desarrollo de una aplicación para detectar y eliminar virus de ordenadores. En el siguiente apartado se presentarán los fundamentos teóricos que sustentan la investigación.

2.2.2. Modelo de prototipos

El Modelo de prototipos, en Ingeniería de software, pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. Este diseño conduce a la construcción de un prototipo, el cual es evaluado por el cliente para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo. Las fases de este tipo de desarrollo son:

- Comunicación
- Plan rápido.

- Modelado, diseño rápido
- Construcción del Prototipo
- Desarrollo, entrega y retroalimentación
- Entrega del desarrollo final

2.2.3. Seguridad informática

La seguridad informática es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático. De todas formas, no existe ninguna técnica que permita asegurar la inviolabilidad de un sistema. Un sistema informático puede ser protegido desde un punto de vista lógico (con el desarrollo de software) o físico (vinculado al mantenimiento eléctrico, por ejemplo). Por otra parte, las amenazas pueden proceder desde programas dañinos que se instalan en la computadora del usuario (como un virus) o llegar por vía remota (los delincuentes que se conectan a Internet e ingresan a distintos sistemas).

En el caso de los virus hay que subrayar que en la actualidad es amplísima la lista de ellos que existen y que pueden vulnerar de manera palpable cualquier equipo o sistema informático. Así, por ejemplo, nos encontramos con los llamados virus residentes que son aquellos que se caracterizan por el hecho de que se hallan ocultos en lo que es la memoria RAM y eso les da la oportunidad de interceptar y de controlar las distintas operaciones que se realizan en el ordenador en cuestión llevando a cabo la infección de programas o carpetas que formen parte fundamental de aquellas.

De la misma forma también están los conocidos virus de acción directa que son aquellos que lo que hacen es ejecutarse rápidamente y extenderse por todo el equipo trayendo consigo el contagio de todo lo que encuentren a su paso.

Los virus cifrados, los de arranque, los del fichero o los sobreescritura son igualmente otros de los peligros contagiosos más importantes que pueden afectar a nuestro ordenador.

Entre las herramientas más usuales de la seguridad informática, se encuentran los programas antivirus, los cortafuegos o firewalls, la encriptación de la información y el uso de contraseñas (passwords).Herramientas todas ellas de gran utilidad como

también lo son los conocidos sistemas de detección de intrusos, también conocidos como anti-spyware. Se trata de programas o aplicaciones gracias a los cuales se puede detectar de manera inmediata lo que son esos programas espías que se encuentran en nuestro sistema informático y que lo que realizan es una recopilación de información del mismo para luego ofrecérsela a un dispositivo externo sin contar con nuestra autorización en ningún momento. Entre este tipo de espías destaca, por ejemplo, Gator.

Un sistema seguro debe ser íntegro (con información modificable sólo por las personas autorizadas), confidencial (los datos tienen que ser legibles únicamente para los usuarios autorizados), irrefutable (el usuario no debe poder negar las acciones que realizó) y tener buena disponibilidad (debe ser estable). De todas formas, como en la mayoría de los ámbitos de la seguridad, lo esencial sigue siendo la capacitación de los usuarios. Una persona que conoce cómo protegerse de las amenazas sabrá utilizar sus recursos de la mejor manera posible para evitar ataques o accidentes.

En otras palabras, puede decirse que la seguridad informática busca garantizar que los recursos de un sistema de información sean utilizados tal como una organización o un usuario lo ha decidido, sin intromisiones.

2.2.4. Virus informático

Es un software que tiene por objetivo de alterar el funcionamiento normal de cualquier tipo de dispositivo informático, sin el permiso o el conocimiento del usuario principalmente para lograr fines maliciosos sobre el dispositivo. Los virus, habitualmente, reemplazan archivos ejecutables por otros infectados con el código de este. Los virus pueden destruir, de manera intencionada, los datos almacenados en una computadora, aunque también existen otros más inofensivos, que solo producen molestias o imprevistos.

Los virus informáticos tienen básicamente la función de propagarse a través de un software, son muy nocivos y algunos contienen además una carga dañina (payload) con distintos objetivos, desde una simple broma hasta realizar daños importantes en los sistemas, o bloquear las redes informáticas generando tráfico inútil. El funcionamiento de un virus informático es conceptualmente simple. Se ejecuta un programa que está

infectado, en la mayoría de las ocasiones, por desconocimiento del usuario. El código del virus queda residente (alojado) en la memoria RAM de la computadora, incluso cuando el programa que lo contenía haya terminado de ejecutar. El virus toma entonces el control de los servicios básicos del sistema operativo, infectando, de manera posterior, archivos ejecutables que sean llamados para su ejecución. Finalmente se añade el código del virus al programa infectado y se graba en el disco, con lo cual el proceso de replicado se completa.

El primer virus atacó a una máquina IBM Serie 360 (y reconocido como tal). Fue llamado Creeper, (ENMS) creado en 1972. Este programa emitía periódicamente en la pantalla el mensaje: «I'm the creeper... catch me if you can!» («¡Soy una enredadera... Atrápame si puedes!»). Para eliminar este problema se creó el primer programa antivirus denominado Reaper (segador).

Sin embargo, el término virus no se adoptaría hasta 1984, pero estos ya existían desde antes. Victor Vyssotsky, Robert Morris Sr. y Doug McIlroy, investigadores de Bell Labs).desarrollaron un juego de ordenador llamado Darwin (del que derivará Core Wars) que consiste en eliminar al programa adversario ocupando toda la RAM.

Existen diversos tipos de virus, varían según su función o la manera en que este se ejecuta en nuestra computadora alterando la actividad de la misma, entre los más conocidos se encuentra:

- Recycler: Consiste en crear un acceso directo de un programa y eliminar su aplicación original, además al infectar un pendrive convierte a toda la información en acceso directo y oculta el original de modo que los archivos no puedan ser vistos, pero con la creación de un archivo batch que modifique los atributos de los archivos contenidos en el pendrive, estos podrían ser recuperados.
- Troyano: Consiste en robar información o alterar el sistema del hardware o en un caso extremo permite que un usuario externo pueda controlar el equipo.

- Bombas lógicas o de tiempo: Son programas que se activan al producirse un acontecimiento determinado. La condición suele ser una fecha (bombas de tiempo), una combinación de teclas, o ciertas condiciones técnicas (bombas lógicas). Si no se produce la condición permanece oculto al usuario.
- Gusano: Tiene la propiedad de duplicarse a sí mismo.
- Hoax: Los hoax no son virus ni tienen capacidad de reproducirse por sí solos. Son mensajes de contenido falso que incitan al usuario a hacer copias y enviarla a sus contactos. Suelen apelar a los sentimientos morales («Ayuda a un niño enfermo de cáncer») o al espíritu de solidaridad («Aviso de un nuevo virus peligrosísimo») y, en cualquier caso, tratan de aprovecharse de la falta de experiencia de los internautas novatos.
- Joke: Al igual que los hoax, no son virus, pero son molestos, un ejemplo: una página pornográfica que se mueve de un lado a otro, y si se le llega a dar a cerrar es posible que salga una ventana que diga error.

Otros tipos por distintas características son los que se relacionan a continuación:

- Virus residentes: La característica principal de estos virus es que se ocultan en la memoria RAM de forma permanente o residente. De este modo, pueden controlar e interceptar todas las operaciones llevadas a cabo por el sistema operativo, infectando todos aquellos ficheros y/o programas que sean ejecutados, abiertos, cerrados, renombrados, copiados. Algunos ejemplos de este tipo de virus son: Randex, CMJ, Meve, MrKlunky.
- Virus de acción directa: Al contrario que los residentes, estos virus no permanecen en memoria. Por tanto, su objetivo prioritario es reproducirse y actuar en el mismo momento de ser ejecutados. Al cumplirse una determinada condición, se activan y buscan los ficheros ubicados dentro de su mismo directorio para contagiarlos.

- Virus de sobreescritura: Estos virus se caracterizan por destruir la información contenida en los ficheros que infectan. Cuando infectan un fichero, escriben dentro de su contenido, haciendo que queden total o parcialmente inservibles.
- Virus de boot (bot_kill) o de arranque: Los términos boot o sector de arranque hacen referencia a una sección muy importante de un disco o unidad de almacenamiento CD, DVD, memorias USB, etc. En ella se guarda la información esencial sobre las características del disco y se encuentra un programa que permite arrancar el ordenador. Este tipo de virus no infecta ficheros, sino los discos que los contienen. Actúan infectando en primer lugar el sector de arranque de los dispositivos de almacenamiento. Cuando un ordenador se pone en marcha con un dispositivo de almacenamiento, el virus de boot infectará a su vez el disco duro.

  Los virus de boot no pueden afectar al ordenador mientras no se intente poner en marcha a este último con un disco infectado. Por tanto, el mejor modo de defenderse contra ellos es proteger los dispositivos de almacenamiento contra escritura y no arrancar nunca el ordenador con uno de estos dispositivos desconocido en el ordenador.
- Virus de enlace o directorio: Los ficheros se ubican en determinadas direcciones (compuestas básicamente por unidad de disco y directorio), que el sistema operativo conoce para poder localizarlos y trabajar con ellos.

  Los virus de enlace o directorio alteran las direcciones que indican donde se almacenan los ficheros. De este modo, al intentar ejecutar un programa (fichero con extensión EXE o COM) infectado por un virus de enlace, lo que se hace en realidad es ejecutar el virus, ya que este habrá modificado la dirección donde se encontraba originalmente el programa, colocándose en su lugar. Una vez producida la infección, resulta imposible localizar y trabajar con los ficheros originales.

- Virus cifrados: Más que un tipo de virus, se trata de una técnica utilizada por algunos de ellos, que a su vez pueden pertenecer a otras clasificaciones. Estos virus se cifran a sí mismos para no ser detectados por los programas antivirus. Para realizar sus actividades, el virus se descifra a sí mismo y, cuando ha finalizado, se vuelve a cifrar.
- Virus polimórficos: Son virus que en cada infección que realizan se cifran de una forma distinta (utilizando diferentes algoritmos y claves de cifrado). De esta forma, generan una elevada cantidad de copias de sí mismos e impiden que los antivirus los localicen a través de la búsqueda de cadenas o firmas, por lo que suelen ser los virus más costosos de detectar.
- Virus multipartitos: Virus muy avanzados, que pueden realizar múltiples infecciones, combinando diferentes técnicas para ello. Su objetivo es cualquier elemento que pueda ser infectado: archivos, programas, macros, discos, etc.

2.2.5 MD5

Es un algoritmo que proporciona un código asociado a un archivo o un texto concreto. De esta forma, a la hora de descargar un determinado archivo, como puede ser un instalador, el código generado por el algoritmo, también llamado hash, viene junto al archivo.

El algoritmo MD5 tiene varios usos además de asegurarnos si un instalador es fiable. El primero de ellos, es que, mediante un programa, también podemos crear el código MD5 de un archivo propio, para que quien haga uso de él pueda comprobar su integridad.

En las instalaciones de firmware proporciona la información referente a la seguridad del archivo y comprueba que la descarga de éste se ha realizado correctamente, y dispongamos del archivo completo y correcto. Esto es de gran utilidad a la hora de instalar un nuevo firmware o sistema operativo en nuestros dispositivos, ya que realizar una instalación de estas características con un archivo dañado o

incompleto, puede dejarnos en ocasiones con un dispositivo inutilizable, o hacernos perder una buena parte de nuestro tiempo.

Otra utilidad que podemos darle al algoritmo MD5 es la de poder comprobar que un texto no haya sido modificado, y haya podido llegar de forma distinta a como era de forma original. Existe software, e incluso páginas web, en las que podemos escribir un texto y que éstas nos devuelven su hash; así, ofreciéndole este dato a nuestro destinatario, éste podrá comprobar si el texto que le hemos enviado no ha sido alterado antes de llegar hasta él.

2.3. Términos fundamentales

Aplicación: Una aplicación (también llamada app) es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático.

Virus: Un virus o virus informáticos es un software que tiene por objetivo de alterar el funcionamiento normal de cualquier tipo de dispositivo informático, sin el permiso o el conocimiento del usuario principalmente para lograr fines maliciosos sobre el dispositivo.

Antivirus: Los antivirus son programas cuyo objetivo es detectar y eliminar virus informáticos. Con el transcurso del tiempo, la aparición de sistemas operativos más avanzados e internet, los antivirus han evolucionado hacia programas más avanzados que además de buscar y detectar virus informáticos consiguen bloquearlos, desinfectar archivos y prevenir una infección de los mismos.

Lenguaje de Programación Python: Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Tkinter: Es un biding de la biblioteca gráfica Tcl/Tk del lenguaje de programación Python que permite la realización de interfaces gráficas de usuario estándar.

CAPÍTULO III
MARCO METODOLÓGICO

3.1 Tipo de investigación

En función del objetivo, el presente trabajo a realizar es un proyecto especial, de acuerdo con el Manual de Trabajos de Grado de Especialización, Maestrías y Tesis Doctorales de la Universidad Pedagógica Experimental Libertador (2016) lo define como: "Trabajos que lleven a creaciones tangibles, susceptibles de ser utilizadas como soluciones a problemas demostrados, o que respondan a necesidades e intereses de tipo cultural. Se incluyen en esta categoría los trabajos de elaboración de libros de texto y de materiales de apoyo educativo, el desarrollo de software, prototipos y de productos tecnológicos en general, así como también los de creación literaria y artística."(pag.22).

3.2. Diseño de la investigación

En cuanto al diseño de la investigación se ha elegido el de campo porque los datos de interés se recogieron en forma directa mediantes los resultados de otras investigaciones y análisis de los antivirus y cualidades de los mismos sumado también a los recogido mediante la observación directa y una lista de cotejo. Según el autor Arias (2012), " la investigación de campo es aquella que consiste en la recolección de todos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variables alguna, es decir, el investigador obtiene la información pero no altera las condiciones existentes. De allí su carates de investigación no experimental." (pag.31)

3.3 Nivel de la Investigación

Con el fin de conocer todas aquellas variables que afectan al desarrollo del sistema, la siguiente investigación es de nivel descriptivo. Arias (2012), define que: "La investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento" (p. 24). En función de esto el estudio del análisis de los datos obtenidos buscando identificar las

características elementales de los antivirus para así darle solución al problema planteado.

3.4. Población y muestra

3.4.1. Población

La población, según Flames (2012), se define como “el conjunto de personas con características afines.”. (p. 24), el mismo autor señala la muestra como “un sub-conjunto representativo de la población que se toma para realizar el estudio.”(p.24). La población del presente trabajo está conformada por 10 expertos que en sus trabajos y vida diaria utilicen un dispositivo con un antivirus instalado.

3.4.2. Muestra

La muestra es la que permite al investigador evaluar la problemática o un fenómeno, ya que esta misma genera datos por medio de los cuales se puede hacer inferencias al mismo problema. Según Arias (2012), “Una muestra es un subconjunto representativo y finito que se extrae de la población accesible.”(p.83). Para la presente investigación se tomará como muestra a los 4 primeros expertos en el área que hayan aceptado responder la entrevista.

3.5. Técnicas e instrumento de recolección de datos

Para Flames (2012), “Las técnicas de recolección de datos son una directriz metodológica que implica el cómo se van a recopilar los datos e informaciones.”(p.26). La utilizada para recopilar los datos fue la observación directa. Esta se define según Méndez (2007) como “el proceso mediante el cual se perciben deliberadamente ciertos rasgos existentes en la realidad por medio de un esquema conceptual previo y con base en ciertos propósitos definidos generalmente por una conjetura que se quiere investigar.”(p.99).

De igual forma, Flames (2012) afirma que “los instrumentos de recolección de datos son recursos metodológicos que implican el con qué se van a recopilar los datos e informaciones.” (p. 26) el instrumento empleado en el presente trabajo es una lista de cotejo. Dicha lista consiste en un listado de aspectos a evaluar, por ejemplo contenidos,

capacidades, habilidades, conductas, entre otros, al lado de los cuales se puede calificar un puntaje, una nota o un concepto.

Es entendido básicamente como un instrumento de verificación. Es decir, actúa como un mecanismo de revisión durante el proceso de enseñanza-aprendizaje de ciertos indicadores prefijados y la revisión de su logro o de la ausencia del mismo.

3.6. Fase metodológica

En el presente proyecto se emplea la metodología de desarrollo prototipo para el desarrollo del presente programa, la aplicación costa de 5 fases

Fase I: Realizar un diagnóstico comparativo entre las características necesarias según los entrevistados mediante una lista de cotejo. En esta fase se realizó el diagnostico comparativo de los requerimientos y características que poseen cada uno de los antivirus preferidos. Para lograr la realización de la tarea se empleó una lista de cotejo.

Fase II: Determinar los requerimientos funcionales y no funcionales en función al diagnóstico anteriormente realizado. Esta fase consto del análisis de los datos obtenidos en la fase anterior. Estos datos serán empleados para determinar los requerimientos y características que debe tener un antivirus para ser funcional.

Fase III: Diseñar el antivirus utilizando el modelo de prototipos. En función de lo obtenido previamente y utilizado las 6 fase que tiene el modelo de prototipado se diseñara el antivirus y en conjunto con la siguiente fase se culminara el diseño y desarrollo.

Fase IV: Desarrollar el antivirus mediante la utilización del lenguaje Python. Esta fase se realizara todo el desarrollo del antivirus utilizando los requerimientos determinados en la fase II y siguiendo el diseño definido en la fase III.

Fase V: Probar el funcionamiento del antivirus mediante las pruebas de caja negra y caja blanca. La última fase consta de poner a prueba el funcionamiento del antivirus para así poder corregir cualquier falla en el mismo. Para completar esta fase se utilizaran las pruebas de caja blanca y caja negra.

CAPÍTULO IV
ANÁLISIS Y RESULTADOS

4.1 Fase I: Realizar un diagnóstico comparativo entre las características necesarias según los entrevistados mediante una lista de cotejo.

En esta fase, primero se procedió a entrevistar a 4 profesionales de distintas áreas que utilicen un software antivirus en los equipos destinados a su trabajo profesional. En cuanto al perfil de los entrevistados para la presente investigación se buscaron personas con conocimientos en el área de computación y de preferencias que lidien y tengan conocimientos del funcionamiento de un antivirus. Además, se buscó personas que los utilicen en el día a día.

Estos requisitos son puestos como parte del perfil que debe tener el usuario ya que se busca obtener información necesaria para conocer los requerimientos de un antivirus y que quejas tienen los usuarios. Si bien respecto a las quejas de los usuarios, se podrían obtener de cualquier persona, pero se necesita tener conocimiento del mismo para saber los requerimientos que deben tener. Los 4 sujetos seleccionados para ser entrevistados fueron aquellos que estaban dispuestos a contestar la encuesta y que a su vez cumplían con todo los requisitos planteados nombrados anteriormente.

Se utilizó un cuestionario de preguntas abiertas con la finalidad de que a través de este, se facilitara el análisis de las características necesarias para realizar un programa que elimine software malicioso que sea de utilidad. A continuación, se presentan los 4 items correspondientes de la entrevista junto con las respuestas de cada individuo.

Item 1: ¿Prefiere un antivirus gratuito o uno pago?

- Sujeto 1: Uno gratuito, hasta ahora me han servido muy bien.
- Sujeto 2: Gratuito, cumplen su función y puedo utilizar el dinero en otros proyectos.

- Sujeto 3: Prefiero un antivirus de pago ya que estos mismos cuentan con una mayor seguridad además de que reciben actualizaciones constantes y soporte.

  Sujeto 4: En mi experiencia puedo decir que se puede utilizar uno gratuito sin preocupaciones ya que si uno conoce como debe proceder a la hora de usar su ordenador no debería tener problemas, pero si un antivirus de pago ofrece más nivel y calidad de seguridad.

Item 2: ¿Cuál es su antivirus de preferencia? ¿Por qué?

- Sujeto 1: AVG. Por recomendación, y porque me ha funcionado bien desde que lo estoy usando.

  Sujeto 2: Avast, porque es gratuito y no genera conflicto con otro software
- Sujeto 3: McAfee debido a que considero el que me va mejor, como mencione tiene una buena seguridad y un buen soporte de su compañía.

  Sujeto 4: Utilizo más de uno pero diría que el que más suelo utilizar es Windows Defender.

Item 3: ¿Usted ha tenido problemas con su antivirus? Si es así, ¿cuáles?

- Sujeto 1: No, hasta ahora ninguno.
- Sujeto 2: Si, no me defiende contra algunas amenazas.
- Sujeto 3: Actualmente no pero antes de adquirir un antivirus de pago tenía ciertos problemas a la hora de desinfección de amenazas siendo que algunas seguían existiendo a pesar de ser supuestamente eliminados y también fallaba mucho a la hora de detectar amenazas.

  Sujeto 4: No he tenido ningún problema con mi antivirus de uso diario.

Item 4: ¿Considera vital que un antivirus no afecte el rendimiento o desempeño normal de los equipos? ¿Por qué?

- Sujeto 1: Si. Porque está entre sus funciones, facilitar el desempeño del equipo.
- Sujeto 2: Sí, es vital porque de ello depende mi uso del ordenador.
- Sujeto 3: Si el que un antivirus no afecte el rendimiento natural del equipo es vital porque la función de un antivirus a mi parecer es proteger al equipo y

mantener su rendimiento normal sería contra producente tener un antivirus que afecte el rendimiento del equipo.

- Sujeto 4: Si un antivirus debe ser usado para protección y mantener el rendimiento óptimo del ordenador ante cualquier factor externo, esto incluye al mismo antivirus.

	AVG		McAfee		Avast		Windows Defender	
	Si	No	Si	No	Si	No	Si	No
Cuenta con una versión paga de mejor calidad	x		x		x			x
La versión gratuita falla durante la detección o eliminación de la amenaza	x		x		x			x
¿Afecta el rendimiento del equipo?		x		x		x		x

Tabla 1. Lista de cotejo
Fuente: Castelli y Pérez (2020)

A partir de las entrevistas realizadas a los sujetos y la tabla de cotejo, podemos determinar las principales preocupaciones y molestias de los usuarios de un antivirus. Primero, se tiene como una de las respuestas notables que el antivirus debe de realizar sus actividades sin que se intervenga con el funcionamiento normal del ordenador. Otro punto notable es con respecto a que un antivirus notifique sus acciones al usuario, esto porque en muchas ocasiones un antivirus elimina programas sin que el usuario esté consciente de ellos y esto puede causar inconvenientes en algunos casos.

Otro aspecto dicho por los usuarios es que un antivirus debe tener un buen mecanismo de detección, lo que quiere decir, que detecte los programas maliciosos de forma efectiva, analizando todo el disco duro del computador del usuario. Por otro lado tenemos en menor medida puntos mencionados como que debería tener una interface simple para que el usuario pueda entenderlo.

4.2 Fase II: Determinar los requerimientos funcionales y no funcionales en función al diagnóstico anteriormente realizado.

Esta fase constó del análisis de los datos obtenidos en la fase anterior. Estos datos serán empleados para determinar los requerimientos y características que debe tener un antivirus para ser funcional. Para que todo desarrollo de un software se lleve a cabo con éxito se deben de identificar los requerimientos funcionales y no funcionales, de esta manera se definen las actividades y los procesos que el sistema debe de realizar. Esto permitirá que las decisiones que se tomen a la hora del desarrollo sean más óptimas debido a que se tienen objetivos a cumplir bien definido.

Por un lado los requerimientos funcionales son las descripciones explícitas del comportamiento que debe tener una solución de software y que información debe manejar, dicho de otra manera los requerimientos funcionales son aquellos que restringen al sistema en cómo reaccionar a entradas, que debe hacer y que no sebe hacer el software y son necesarios para que funcione el sistema correctamente

Por otra parte los requerimientos no funcionales describen otras prestaciones, características y limitaciones que debe tener el sistema para alcanzar el éxito. Los requerimientos no funcionales engloban características como rendimiento, facilidad de uso, presupuestos, tiempo de entrega, documentación, seguridad. Entre estos requerimientos, uno de los más importantes para un programa es la facilidad de uso.

Los requerimientos obtenidos de la fase 1 son:

Funcionales:

1. Ser capaz de detectar virus, gusanos y troyanos.
2. Eliminar la amenaza.
3. Tener un mecanismo para escanear los archivos ejecutables del ordenador. y detectar cual está infectado con un virus, gusano o troyano.

No funcionales:

4. Tener una interface de simple entendimiento.
5. Informarle al usuario sobre las acciones del antivirus.

Utilizando los requerimientos anteriormente señalados, se realizó el siguiente cuadro:

Requerimientos	Funcional	No funcional
Ser capaz de detectar virus, gusanos y troyanos.	X	
Eliminar la amenaza	X	
Tener un mecanismo para escanear los archivos ejecutables del ordenador y detectar cual está infectado con un virus, gusano o troyano.	X	
Tener una interface de simple entendimiento		X
Informarle al usuario sobre las acciones del antivirus		X

Tabla 2. Requerimientos funcionales y no funcionales

Fuente: Castelli y Pérez (2020)

A partir de la definición de estos requerimientos, se puede tener una idea clara de la finalidad del programa y de cuáles son los aspectos necesarios para que funcione como es debido. Un antivirus básico debe de contar con los requerimientos anteriormente mencionados para que cumpla con su objetivo. Además, realizado este análisis, es posible esquematizar el modelado y diseño rápido en la posterior etapa.

4.3 Fase III: Diseñar el antivirus utilizando el modelo de prototipos

En función de lo obtenido en la fase anterior, y utilizando las 6 etapas que tiene el modelo de prototipos, se es posible programar rápidamente un prototipo del software y ver cómo funciona. El objetivo de este modelo es elaborar un prototipo del programa final y mostrar su funcionamiento al cliente. De esta forma, se diseñará el antivirus de la realizando lo siguiente:

4.3.1 Etapa 1: recolección y refinamientos de requisitos

Esta etapa fue la primera en ser completada. En ella, encuentra englobada toda la recolección de información mediante las entrevistas y lista de cotejo realizadas en la Fase I. La información obtenida permitió analizar los aspectos más importantes del sistema: realizar las actividades sin que se intervenga con el funcionamiento normal del ordenador, notificar las acciones al usuario y tener un buen mecanismo de detección.

Además, utilizando esta información se realizó la especificación de los requerimientos del sistema de la Fase II: ser capaz de detectar virus, gusanos y troyanos, eliminar la amenaza, tener un mecanismo para escanear los archivos ejecutables del ordenador y detectar cual está infectado con un virus, gusano o troyano, tener una interface de simple entendimiento e informarle al usuario sobre las acciones del antivirus.

4.3.2 Etapa 2: modelado y diseño rápido

Para construir un prototipo, lo primero que se debe hacer es realizar un modelo del sistema, a partir de los requisitos que ya se conocen En este caso no es necesario realizar una definición completa de los requisitos, pero sí es conveniente determinar al menos las áreas donde será necesaria una definición posterior más detallada.

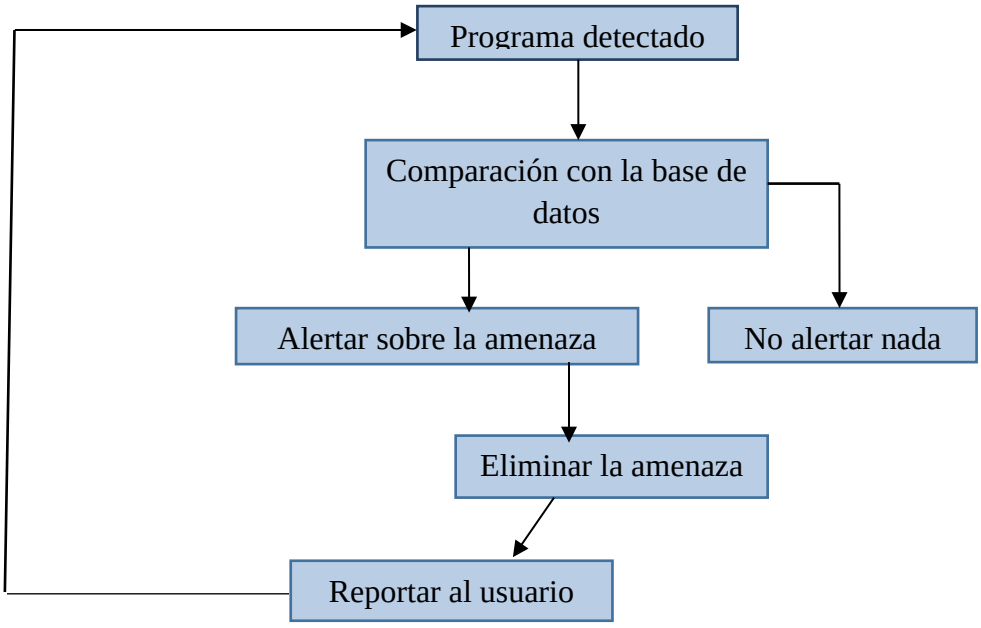

Figura 1. Diseño rápido
Fuente: Castelli y Pérez (2020)

4.3.3 Etapa 3: Construcción del prototipo

En función de lo obtenido previamente, y utilizando las 6 etapas que tiene el modelo de prototipos, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Este modelo se utilizan para dar al usuario una vista preliminar de parte del software este modelo se compone de 6 etapas para su culminación en fusión de las mimas se diseñará el antivirus de la siguiente forma:

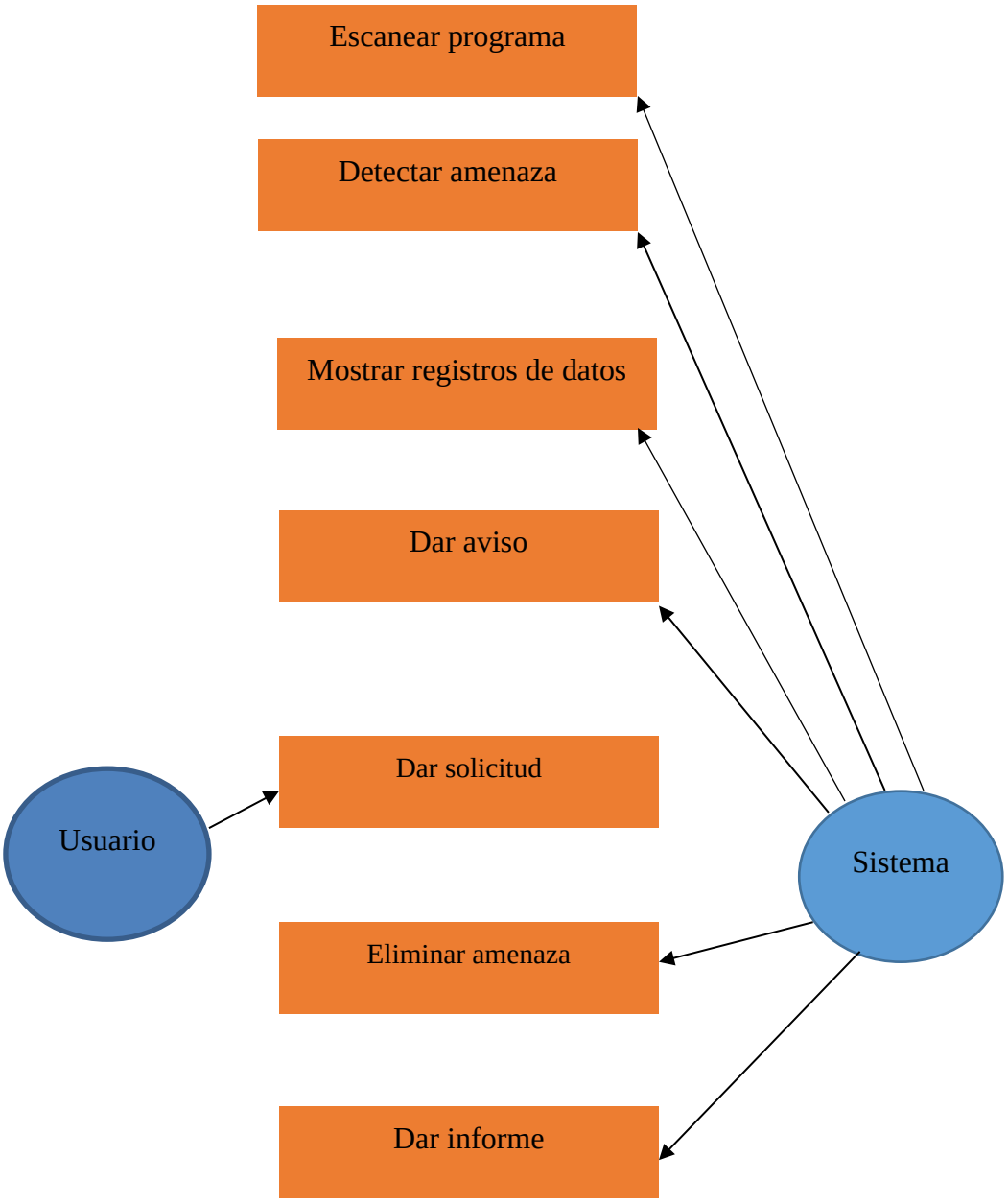

Figura 2. Diagrama de uso-diseño rápido
Fuente: Castelli y Pérez (2020)

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). En el presente diagrama se muestra como es la interacción del usuario (cliente) y el programa siendo que el mismo solo le expresas las solicitudes que desea al programa y este las ejecuta.

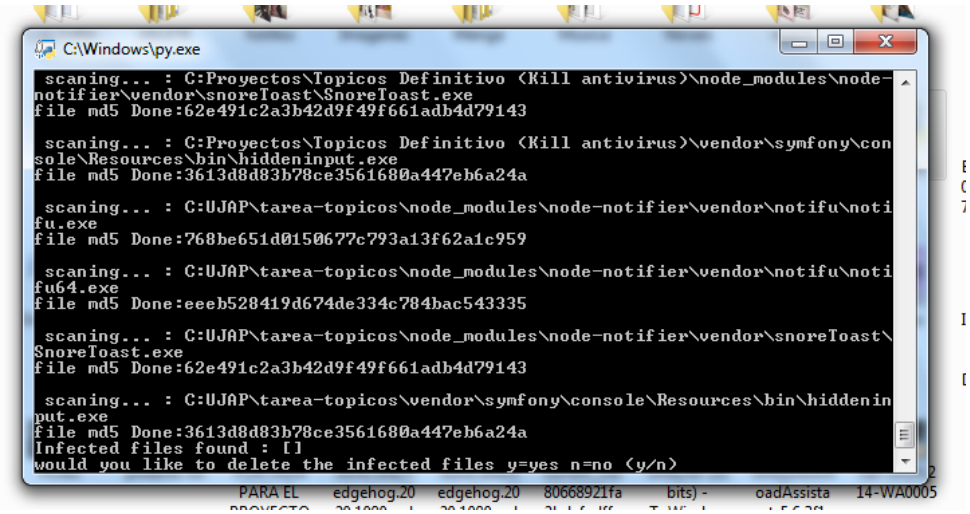

Figura 3. Prueba de corrida del prototipo

Se presenta el funcionamiento del primer prototipo de antivirus realizado este solo consistía en un escaneo superficial e imprimía todo por consola, no tenía interfaz gráfica y la respuesta se daba solo para borrar.

```
print("start scan files....")
def counts():
        for x in range(5):
                print(x+1)
                time.sleep(1)
counts()


def scan():
        infected_list=[]
        for f in file_list:
                virus_def=open("viruses.txt","r")
                file_not_read=False
                print("\n scaning... : {}".format(f))
                hasher=hashlib.md5()
                try:
                        with open(f,"rb") as file:
                                try:
                                        buf=file.read()
                                        file_not_read=True
                                        hasher.update(buf)
                                        file_hashed=hasher.hexdigest()
                                        print("file md5 Done:{}".format(file_hashed))
                                        for line in virus_def:
                                                if file_hashed== line.strip():
                                                        print("Malware Detected --> file name: {}".format(f))
                                                        infected_list.append(f)
                                                else:
                                                        pass
                                        except Exception as e:
                                                print(" could not read the file Error: {}".format(e))
                        except:
                                pass
        print("Infected files found : {}".format(infected_list))
        deleteOrnot=str(input("would you like to delete the infected files y=yes n=no (y/n)"))
        if deleteOrnot.upper()=='Y':
                for infected in infected_list:
                        os.remove(infected)
                        print("file removed : {}".format(infected))
        else:
                print(" See You ...")
```

Figura 4. Código del prototipo

Se presenta el código del primer prototipo de antivirus realizado este solo consistía en un escaneo superficial e imprimía todo por consola, no tenía interfaz gráfica y la respuesta se daba solo para borrar.

4.3.4 Etapa 4: Desarrollo, evaluación del prototipo por el cliente

En esta fase, como fue mencionado anteriormente, inicia el desarrollo del prototipo, que en este caso se desarrolló con el lenguaje de programación Python. Una vez terminado, se le enseñó el prototipo al cliente, quien lo evaluó y dio su opinión. La evaluación dada fue primero con respecto a la interfaz, que es muy importante debido a que los usuarios promedio no se sienten cómodo trabajando por consola.

Además, que los resultados salieran en consola no era satisfactorio y se decidió imprimir los resultados en un área de texto en la interfaz. Por otro lado, en vez de dar una respuesta con una letra se creó un botón de esa manera el usuario solo tiene que presionarlo para dar su petición al programa. Por último, se incluyó un botón para cerrar el programa con esta opinión se tomara para seguir en el desarrollo del programa.

También se le fue agregada una ventana principal junto a la fecha del último escaneo realizado y un botón que redirige a otra ventana donde se incluyeron las siguientes opciones: realizar escaneo completo del disco duro, ver el historial de amenazas borradas, la base de datos en MD5 del programa, el historial de los escaneos realizados y la opción de ver solo las amenazas detectadas.

4.3.5 Etapa 5: refinamiento del prototipo

En función de todas las opiniones y críticas dadas por el cliente se refinó y adaptó el prototipo para que cumplan con las necesidades del cliente. Esta etapa puede ser cíclica, ya que muchas veces una vez entregado el prototipo refinado, aún puede ser modificado y refinado una vez más, una vez que el prototipo este totalmente refinado se dará por completada esta etapa.

El refinamiento tuvo como resultado lo siguiente:

Figura 5. Pantalla principal

Muestra la pantalla principal del programa junto con la última vez que se realizó un escaneo del disco duro.

Figura 6. Ventana de acciones

Muestra la ventana de acciones.

Figura 7. Análisis finalizado

Muestra los archivos analizados del disco duro.

Figura 8. Amenaza mostrada

Muestra el archivo malicioso analizado mediante el escaneo

Figura 9. Base de datos en MD5

Muestra la base de datos

Figura 10. Historial de amenazas detectadas

Muestra todas las amenazas detectadas en cada escaneo del disco duro.

Figura 11. Historial de escaneos realizados
Muestra la fecha y hora exacta en la que se realizó un escaneo del sistema.

4.3.6 Etapa 6: entrega del producto final

Como fase final se encuentra por su puesto la entrega del producto**.** Esta etapa va en conjunto con la fase final de la investigación. Como producto final, el programa es capaz de detectar gusanos, troyanos y virus almacenados en la base de datos como MD5 y eliminarlos del disco duro. Se realizaron 30 escaneos de prueba, en los cuales se encontraron satisfactoriamente 9 de las amenazas almacenadas en la base de datos. Si la amenaza no se encuentra agregada en la base de datos, el antivirus no reconoce el MD5 como un archivo infectado. Además de que muestra los historiales de análisis y amenazas detectadas por el programa. Cuenta con una interfaz simple, pero que es fácil de entender.

El producto final tuvo como resultado lo siguiente:

Figura 12. Pantalla principal final

Muestra la pantalla principal del programa junto con la última vez que se realizó un escaneo del disco duro.

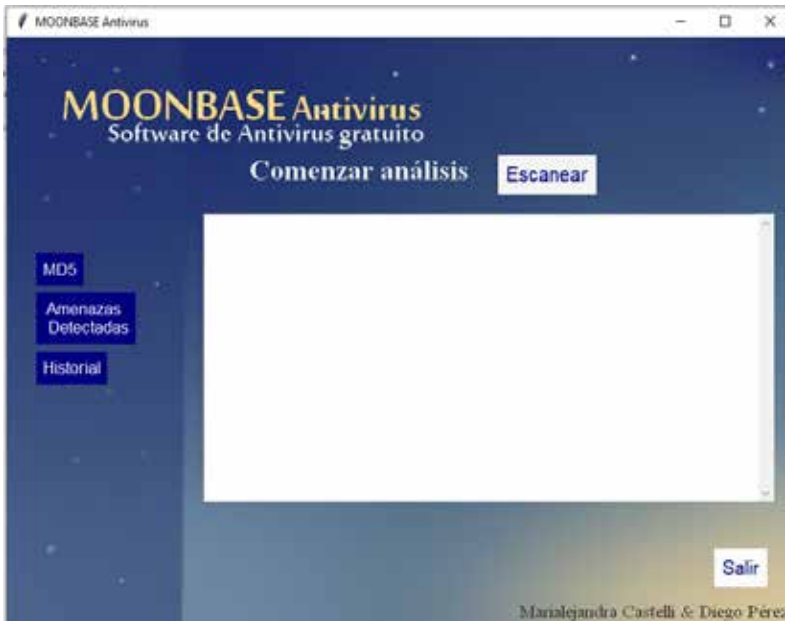

Figura 13. Ventana de acciones final

Muestra la ventana de acciones.

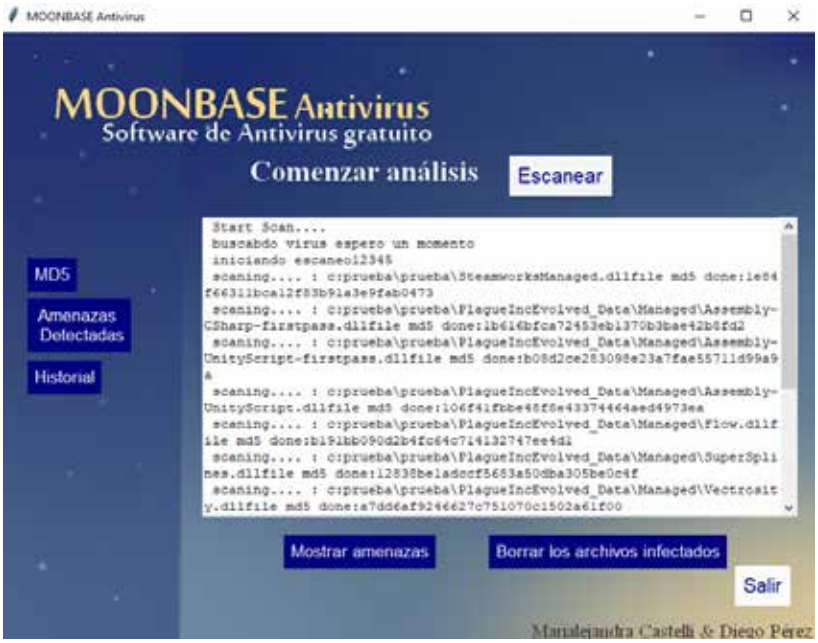

Figura 14. Análisis finalizado final
Muestra los archivos analizados del disco duro.

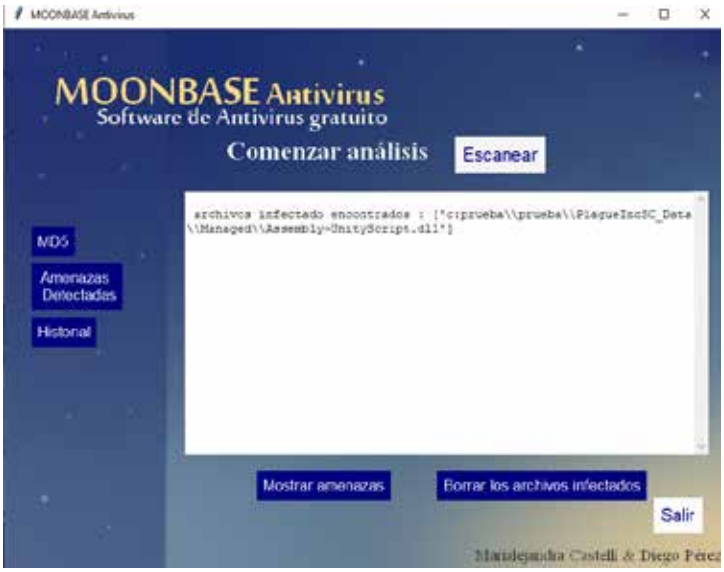

Figura 15. Amenaza mostrada final
Muestra el archivo malicioso analizado mediante el escaneo

Figura 16. Base de datos en MD5 final
Muestra la base de datos

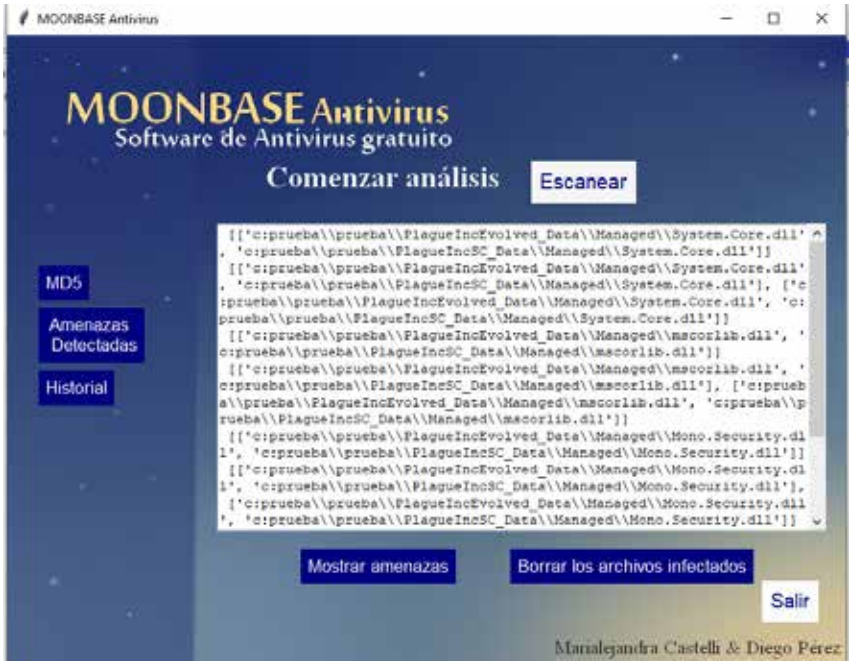

Figura 17. Historial de amenazas detectadas final
Muestra todas las amenazas detectadas en cada escaneo del disco duro.

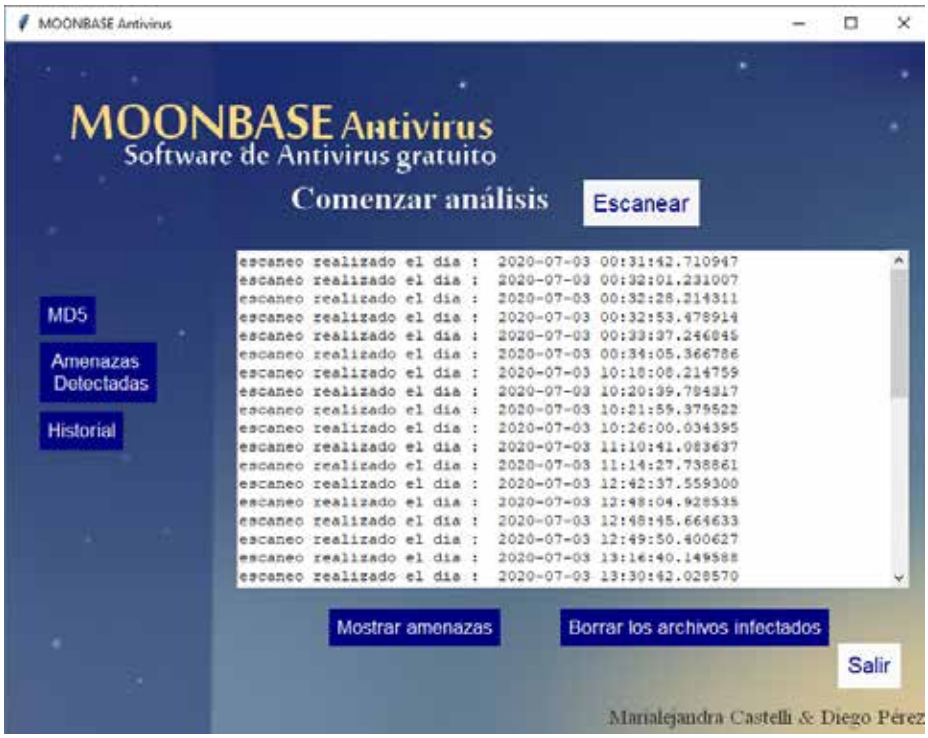

Figura 18. Historial de escaneos realizados final
Muestra la fecha y hora exacta en la que se realizó un escaneo del sistema

4.4 Fase IV: Desarrollar el antivirus mediante la utilización del lenguaje Python

Para el proceso de desarrollo del software, se realizó la implementación del sistema donde se programan los requisitos especificados y tomando como base el prototipo realizado en la fase anterior. Para la realización del mismo se escogió el lenguaje de programación Python, debido a la variedad de librerías que son de suma importancia para el mismo.

Se inició por el desarrollo del método de detección de programas maliciosos para este método se tomó el enfoque de detectar los virus mediante el uso de MD5. El MD5 (Message Digest Algorithm 5) es un algoritmo que se utiliza como una función de codificación o huella digital de un archivo. De esta forma, a la hora de descargar un determinado archivo como puede ser un instalador, el código generado por el algoritmo, también llamado hash, viene “unido” al archivo.

Lo segundo que se desarrolló, fue un método de eliminación y registro de amenazas. Una vez concluido el primer método de detección y aprendizaje se creó un método con los valores de los MD5 que el programa considere malicioso para guardarlo en la base de datos, luego de esto el método de eliminación se efectuará para cuando el programa detecte un programa malicioso le de la información la notificación al usuario y este mismo decida si desea eliminarlo, de desear eliminarlo el programa lo borrará.

Por otro lado, la interfaz gráfica se realizó utilizando la biblioteca gráfica Tcl/Tk de Python, Tkinter, que proporciona la posibilidad de realizar la interfaz gráfica de usuario estándar del lenguaje Python. A través de esta, se implementaron las ventanas, botones y áreas de texto necesarias para iniciar el proceso de escaneo y visualizar los resultados del mismo.

4.5 Fase V: Probar el funcionamiento del antivirus mediante las pruebas de caja negra y caja blanca

Ya con el antivirus terminado se procedió a probar su funcionamiento con las pruebas de caja negra y caja blanca. Las Pruebas de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

En las pruebas de caja negra, se enfoca solamente en las entradas y salidas del sistema, sin darle importancia en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, nos basamos en los requerimientos de software y especificaciones funcionales.

Para la prueba de caja negra se quería verificar que el antivirus reciba correctamente la petición del usuario las condiciones de la entrada son las siguientes:

Condiciones	Correctas	Incorrectas
N	N= bt eliminar, bt salir	N<> (bt eliminar , bt salir)

Tabla 3. Caja Negra-Entradas
Fuente: Castelli y Pérez (2020)

Las entradas correctas son cuando se presionan los botones de eliminar o salir, ya que estas mismas son las respuestas que recibe el programa como peticiones, por ende toda aquella respuesta que no sea estas arrojara un error. Sabiendo esto se procedió a probar 5 entradas 2 correctas y 3 incorrectas.

Clases	Entrada	Salida
Correcta	N=bt salir	Error
Correcta	N=bt eliminar	Se elimina los archivos maliciosos
Incorrecta	N=y	Error
Incorrecta	N=s	Error
Incorrecta	N=1	Error

Tabla 4. Caja Negra-Prueba de entradas
Fuente: Castelli y Pérez (2020)

Gracias a esta prueba se pudo reconocer que la aplicación tenía un defecto al recibir la entrada correcta “bt salir” debido a la aplicación debería cerrarse una vez reciba esta respuesta al no ser eso se pudo identificar la falla para ser corregida. Con las otras entradas no existieron inconvenientes. La entrada “bt eliminar” si dio la salida que se esperaba y con el resto de entradas erróneas efectivamente dio error ya que estas mismas están fuera de los limites.

Una vez corregido los errores se procedió a re aplicar las pruebas con los mismos parámetros:

Clases	Entrada	Salida
Correcta	N=bt salir	Se cierra el programa
Correcta	N=bt eliminar	Se elimina los archivos maliciosos
incorrecta	N=y	Ninguna
Incorrecta	N=s	Ninguna
Incorrecta	N=1	Ninguna

Tabla 5. Caja Negra-Resultados
Fuente: Castelli y Pérez (2020)

En estas pruebas, se le implementaron modificaciones ya la entrada “bt salir” si cierra el programa además que se alteró el código para que cualquier otra respuesta que del usuario el programa no caiga en error y solo quede en espera de una entrada valida.

Siguiendo con las pruebas se realizara las pruebas de caja blanca. Estas se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente se escogen distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados. El cometido de estas pruebas es comprobar los flujos de ejecución dentro de cada unidad (función, clase, módulo, etc.) pero también pueden probar los flujos entre unidades durante la integración, e incluso entre subsistemas, durante las pruebas de sistema.

Para el presente trabajo se diseñara las pruebas de caja blanca usando el método de pruebas de caminos básicos. A continuación el grafo asociado al programa general:

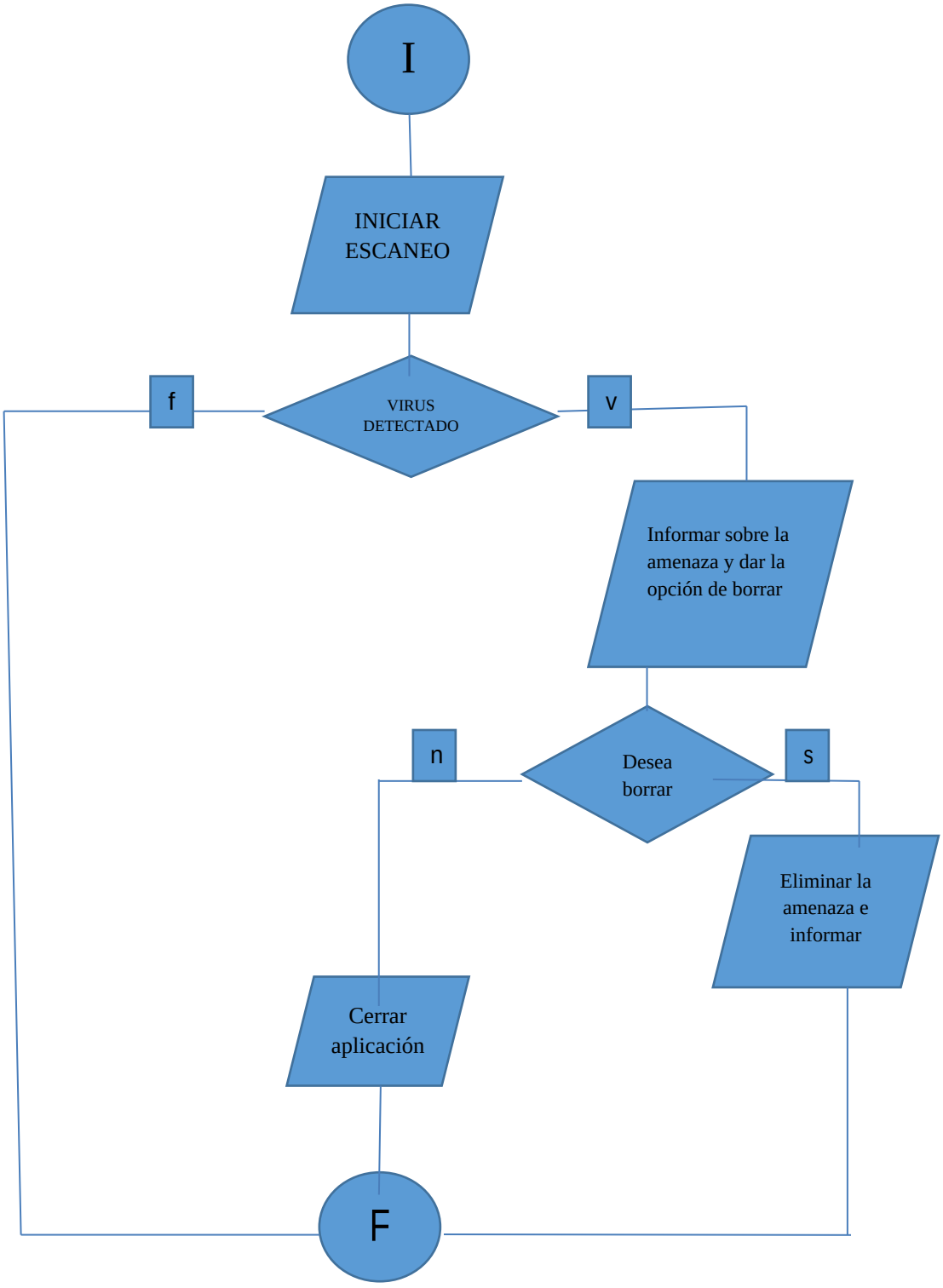

Figura 19. Grafo del programa
Fuente: Castelli y Pérez (2020)

El presente diagrama consta de 2 nodos predicados y por ende, una complejidad ciclomática de 3 y estos resultarán en 3 caminos de solución los cuales son:

Camino	**Recorrido**
1	1,2,3,8
2	1,2,3,4,5,6,8
3	1,2,3,4,5,7,8

Tabla 6. Caja Blanca- Recorridos
Fuente: Castelli y Pérez (2020)

En función de estos caminos procederemos a darle entradas para comprobar si el sistema se comporta como debería:

Camino	**Recorrido**	**Entrada**	**Salida**
1	1,2,3,8	Vacío	Ninguna
2	1,2,3,4,5,6,8	Programa maliciosos	Programa eliminado
3	1,2,3,4,5,7,8	Programas malicioso	Programa no eliminado

Tabla 7. Caja Blanca- Recorridos y entradas
Fuente: Castelli y Pérez (2020)

Como se comprobó en el caso del primer camino que es cuando el programa no detecta ninguna amenaza simplemente no obtiene una entrada y por ende no da una salida, para los siguientes dos caminos la aplicación también se comportó como debería ya que arrojo la respuesta y siguió el camino de ejecución a la perfección. Demostrando así que para esta prueba no se encontraron errores de funcionamiento.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

Con la finalización del desarrollo de una aplicación para detectar y eliminar virus de ordenadores, se llegaron a las siguientes conclusiones:

1. Se realizó un antivirus que cuenta con detección y eliminación de virus, gusanos y troyanos en archivos ejecutables. Que además cuenta con la opción de ver el historial de escaneos realizados y opción para ver la base de datos de los códigos MD5.
2. Un software que tenga como finalidad detectar y eliminar programas maliciosos debe tener un método para analizar y tener acceso a los ejecutables instalados en el ordenador, de lo contrario no es posible extraer su código MD5 para comprobar su fiabilidad, ni su eliminación en caso de que sea peligroso.
3. La interfaz de usuario es fundamental para un programa dirigido al público. Un software de antivirus debe reflejar confianza y seguridad a los usuarios a través de una interfaz agradable y fácil de usar.

En relación a las fases metodológicas, se pudo concluir que:

1. Los antivirus gratuitos comparten ciertas características criticadas por los usuarios y recomendamos que antes de adquirir un antivirus se informen al respecto.
2. Un antivirus debe ser capaz de detectar las amenazas al ordenador y, para poder eliminarla sin problemas, necesita contar con una gran base de datos para identificar los programas maliciosos.
3. La metodología de prototipos fluye con el diseño del antivirus, debido a que el desarrollo del mismo pasó por muchos modelos antes de llegar al producto final. El uso de esta metodología es efectivo para proyectos que involucren un desarrollo de auto mejoramiento en cada modelo.

4. Se puede desarrollar exitosamente un antivirus con el lenguaje de programación Python, ya que cuenta con la versatilidad y librerías apropiadas para ellos.
5. Es vital para cualquier programa probar su funcionamiento antes de ejecutarlo debido a que siempre existirá la posibilidad de encontrar fallas dentro del mismo.

Para mantener la aplicación, hacer actualizaciones del programa o futuras investigaciones al respecto se recomienda:

1. La investigación y uso de otras librerías para interfaz. A través de Tkinter es posible realizar una interfaz funcional y ordenada necesaria para el programa. Sin embargo, estas interfaces son estándar y si se desea agregar más cosas se necesita otra librería.
2. Informarse y mantener actualizada la base de datos de virus del sistema. Al mantener la base de datos actualizada, se asegura que el programa detecte las nuevas amenazas que surjan e intenten poner en riesgo la seguridad del ordenador.
3. Realizar pruebas de control para asegurarse que su funcionamiento siga siendo el correcto y no detecte falsos positivos.

REFERENCIAS

Bibliográficas

Arias, F. (2012). **El proyecto de investigación: introducción a la metodología científica**. Caracas: Episteme. 6ta. Edición.

Supo, J. (2015). **Cómo empezar una tesis- Tu proyecto de investigación en un solo día.** Arequipa: BIOESTADISTICO EIRL.1era. Edición.

Rodríguez, J., Oduber, J. y Mora, E. (29 de junio de 2017). **Actividades rutinarias y cibervictimización en Venezuela.** *Revista Latinoamericana de Estudios de Seguridad, (20)*, p. 63-79.

Ruano-Ordás, D. (2016). **Resumen de tesis: Modelo para la optimización de la ejecución de filtros anti-spam.** *Revista Iberoamericana de Inteligencia Artificial*, (*19*), p. 1-4.

Islam, H. (2018). **An Anti-Malware Product Test Orchestration Solution for Multiple Pluggable Environments (tesis de maestría).** Universidad de Turku, Finlandia.

Bavaresco, A. (2013). **Proceso metodológico en la investigación (Cómo hacer un diseño de investigación).** Maraciabo: Imprenta Internacional, CA. 6ta. edición.

Russel, S. y Norvig, P. (2009). **Inteligencia Artificial: Un Enfoque Moderno.** Englewood Cliffs: Prentice Hall. 3era. Edición.

Universidad Pedagógica Experimental Libertador. (2006). **Manual de Trabajos de Grado de Especialización, Maestrías y Tesis Doctorales.** Caracas: FEDUPEL. 3era. Edición.

Flames, A. (2012). **Trabajo De Grado Cualitativo Y Cuantitativo.** Caracas: IPASME. 3era. Edición.

Méndez, C. (2007). **Metodología, Diseño Y Desarrollo Del Proceso De Investigación.** Colombia: McGraw Hill Interamericana, SA.

Electrónicas

Desarrollo Web (2003). **¿Qué es Python?**. Recuperado de: https://desarrolloweb.com/articulos/1325.php.

Laura, R. y Tumi, E. (2019, 7 de agosto). **Sistema antivirus multiplataforma en tiempo real usando técnicas heurísticas y proactivas.** *Revista de Investigación Ciencia, Tecnología y Desarrollo.* Recuperado de https://revistas.upeu.edu.pe/index.php/RICTD/issue/view/49

Rivera, G. (2013). **Malware y algo más.** U@CSIS. Recuperado de http://investigacionsis.fuac.edu.co/html/RepositorioOJS/ojsfuac/ojs/index.php/UACSIS/issue/view/1

Martínez, I. (2015). **Qué es MD5, cómo funciona y para qué se usa.** Recuperado de: https://rootear.com/seguridad/md5-como-funciona-usos

Roberts, J. (2012). **Hashes.** Recuperado de: https://virusshare.com/

González, G. (2015) **¿Cómo funcionan los antivirus?** Recuperado de:https://blogthinkbig.com/como-funcionan-los-antivirus#:~:text=Los%20antivirus%20funcionan%20en%20segundo,el%20ordenador%20donde%20est%C3%A1n%20instalados.&text=EXE%2C%20este%20no%20se%20abre,se%20conozcan%20hasta%20la%20fecha.

ANEXOS

a. Anexo- Sujeto 1

Entrevista para el Trabajo de Grado: Desarrollo de una aplicación para detectar y eliminar virus de ordenadores.

Realizado por: Marialejandra Castelli y Diego Pérez
Nombre completo: Thais Carolina Alamo Matos
Profesión: Profesora

1. ¿Prefiere un antivirus gratuito o uno pago?
   Gratuito, hasta ahora me han servido muy bien.

2. ¿Cuál es su antivirus de preferencia? ¿Por qué?

   AVG.

   Por recomendación, y porque me ha funcionado bien desde que lo estoy usando

3. ¿Usted ha tenido problemas con su antivirus? Si es así, ¿cuáles?
   No, hasta ahora ninguno.

4. ¿Considera vital que un antivirus no afecte el rendimiento o desempeño normal de los equipos? ¿Por qué?
   Si.
   Porque está entre sus funciones, facilitar el desempeño del equipo.

b. Anexo- Sujeto 2

Entrevista para el Trabajo de Grado: Desarrollo de una aplicación para detectar y eliminar virus de ordenadores.

Realizado por: Marialejandra Castelli y Diego Pérez
Nombre completo: María Tapizquent
Profesión: Lcda en Química

1. ¿Prefiere un antivirus gratuito o uno pago?
   Gratuito, cumplen su función y puedo utilizar el dinero en otros proyectos.

2. ¿Cuál es su antivirus de preferencia? ¿Por qué?
   Avast, porque es gratuito y no genera conflicto con otro software

3. ¿Usted ha tenido problemas con su antivirus? Si es así, ¿cuáles?
   Si, no me defiende contra algunas amenazas

4. ¿Considera vital que un antivirus no afecte el rendimiento o desempeño normal de los equipos? ¿Por qué?
   Sí, es vital porque de ello depende mi uso del ordenador

c. Anexo- Sujeto 3

Entrevista para el Trabajo de Grado: Desarrollo de una aplicación para detectar y eliminar virus de ordenadores.

Realizado por: Marialejandra Castelli y Diego Pérez
Nombre completo: Xavier aguilera
Profesión: programador independiente

1. ¿Prefiere un antivirus gratuito o uno pago?
   Prefiero un antivirus de pago ya que estos mismos cuentan con una mayor seguridad además de que reciben actualizaciones constantes y soporte

2. ¿Cuál es su antivirus de preferencia? ¿Por qué?
   McAfee debido a que considero el que me va mejor, como mencione tiene una buena seguridad y un buen soporte de su compañía

3. ¿Usted ha tenido problemas con su antivirus? Si es así, ¿cuáles?
   Actualmente no pero antes de adquirir un antivirus de pago tenía ciertos problemas a la hora de desinfección de amenazas siendo que algunas seguían existiendo a pesar de ser supuestamente eliminados y también fallaba mucho a la hora de detectar amenazas

4. ¿Considera vital que un antivirus no afecte el rendimiento o desempeño normal de los equipos? ¿Por qué?
   Si el que un antivirus no afecte el rendimiento natural del equipo es vital porque la función de un antivirus a mi parecer es proteger al equipo y mantener su rendimiento normal sería contra producente tener un antivirus que afecte el rendimiento del equipo.

d. Anexo- Sujeto 4

Entrevista para el Trabajo de Grado: Desarrollo de una aplicación para detectar y eliminar virus de ordenadores.

Realizado por: Marialejandra Castelli y Diego Pérez
Nombre completo: Octavio Pérez
Profesión: computista

1. ¿Prefiere un antivirus gratuito o uno pago?
   En mi experiencia puedo decir que se puede utilizar uno gratuito sin preocupaciones ya que si uno conoce como debe proceder a la hora de usar su ordenador no debería tener problemas, pero si un antivirus de pago ofrece más nivel y calidad de seguridad.

2. ¿Cuál es su antivirus de preferencia? ¿Por qué?
   Utilizo más de uno pero diría que el que más suelo utilizar es Windows Defender.

3. ¿Usted ha tenido problemas con su antivirus? Si es así, ¿cuáles?
   No he tenido ningún problema con mi antivirus de uso diario.

4. ¿Considera vital que un antivirus no afecte el rendimiento o desempeño normal de los equipos? ¿Por qué?
   Si un antivirus debe ser usado para protección y mantener el rendimiento óptimo del ordenador ante cualquier factor externo, esto incluye al mismo antivirus.