Desarrollo de una aplicación para la administración de la evidencia digital

Development of an application for the administration of digital evidence

Sánchez Calderón, Doris Milena[1], Tamayo Peñuela, Leidy Johana[1] y Reyes Álvarez, Marcos Fernando[2]
Fundación Universitaria de San Gil - UNISANGIL, Facultad de Ciencias Naturales e Ingeniería
Programa Ingeniería de Sistemas
San Gil, Colombia

dorissanchez@unisangil.edu.co
leidytamayo@unisangil.edu.co
mreyes@unisangil.edu.co

Fecha de recepción: 4 de noviembre de 2016
Fecha de aceptación: 16 de abril de 2018

Resumen — Este artículo de investigación presenta el desarrollo de una aplicación para la administración de la evidencia digital. En la actualidad, la sociedad vive diversos tipos de problemáticas en cuanto a la seguridad de la información y en relación con las tecnologías de la información y las comunicaciones, lo cual ha generado que se desarrollen herramientas para combatir distintos delitos informáticos; por esta razón se hace necesario detectar evidencias de la vulnerabilidad de la privacidad de los archivos, documentos y demás datos particulares. Este tipo de herramientas ha llevado a que los ciberdelincuentes perfeccionen sus habilidades para adulterar archivos en un sistema logrando que sus delitos electrónicos sean difíciles de detectar y contrarrestar; acarreando como consecuencia una gran falta de conocimiento, inversión en la implementación de tecnologías de la información. La construcción de un software con un gestor de base de datos SQLite, pretende dar solución al problema que está relacionado con las actividades sobre la privacidad de los datos, mediante la consulta y el análisis de malware, análisis de metadatos, elaboración de reportes, así mismo la validación y seguridad de los mismos. Aunque hay muchos usuarios de este tipo de herramientas, no son suficientes para mitigar la posibilidad de que se presenten ataques como a bases de datos, escaneo de redes, entre otros.

Palabras clave— Tecnologías de la información y comunicación, SQLite, software, gestor, malware, metadatos.

Abstract **-** This research paper presents the development of an application for the management of digital evidence. Nowadays society lives different types of issues concerning information security and ICTs –Information and Communication Technologies-, which has generated the development of tools to battle different kinds of cyber-crimes, because of that, it is necessary to detect evidence of files privacy vulnerability, documents and other private data. This type of tool has led to cyber-criminals to master their skills to alter files on a system, making their electronic crimes difficult to avoid and counter, having as result a lack of knowledge and investment on the implementation of information technology. Making a software with a SQLite database manager aims to solve the problems related to activities on data privacy through consultation and malware analysis, analysis of metadata, preparing reports and their own security validation. Although there are many users of these tools, they are still not enough to mitigate the possibility of attacks to databases, network scanning, etc.

Keywords - ICTs –Information and Communication Technologies, SQLite, software, manager, malware, metadata.

I. INTRODUCCIÓN

Las comunicaciones y la informática han traído grandes cambios al mundo, que han impactado a la sociedad, entre ellos al sistema de administración de justicia como escenario idóneo para la solución de controversias como la proliferación de delitos informáticos, lo cual genera. conciencia en la aplicación de metodologías de seguridad informática.

La informática forense consiste en la investigación de los sistemas de información con el fin de detectar evidencias de la vulneración de los mismos y es utilizada para garantizar la efectividad de las políticas de seguridad y la protección tanto de la información como de las tecnologías, que facilitan la gestión de la información [1] [3].

Teniendo en cuenta lo anterior, se planteó una alternativa de solución para mitigar la problemática de carencia de una guía de desarrollo que sirva como herramienta, donde se encuentre paso a paso los procesos adecuados de la investigación forense, uso de herramientas gratuitas, donde se explique el manejo de estas y a su vez desarrollar una aplicación para la realización de análisis forense usando como modelo la herramienta Kali Linux [2], técnica informática que permite reconstruir lo que ha sucedido en un sistema tras un incidente de seguridad y donde se puede

[1] Ingeniero de Sistemas, UNISANGIL
[2] Ingeniero de Sistemas, UNISANGIL. Especialista en Seguridad Informática, Universidad Pontificia Bolivariana Seccional Bucaramanga. Coordinador Semillero en Seguridad Informática – SIGSU, UNISANGIL.

determinar quién, desde dónde, cómo, cuándo y qué acciones ha llevado a cabo un intruso en los sistemas afectados.

Por tal motivo, el proyecto surgió con la intención de cubrir dichas necesidades mediante un aplicativo cómodo en su uso, amigable en su entorno, de bajo en costo y práctico para que el usuario pueda llevar un registro de los análisis de sus casos, archivos y reportes de forma ordenada y precisa.

II. DESARROLLO DE LA APLICACIÓN

El auge de las comunicaciones y la informática forense han aportado herramientas para motivar el uso de los medios electrónicos. El 5 de enero de 2009 el Congreso de la República promulga la Ley 1273 denominada "De la Protección de la información y de los datos" [4]. Esta ley es presentada integralmente para los sistemas que utilizan tecnologías de la información y las comunicaciones. Por ello han surgido herramientas que permiten al usuario resolver inconvenientes relacionadas con el delito informático, por medio de la herramienta Kali Linux en modo gráfico [2], se puedan realizar procesos informáticos forenses como análisis de vulnerabilidad, ataques a base de datos, escaneo de redes, entre muchos otros y de forma intuitiva y fácil.

Kali Linux es una distribución de Linux avanzada para pruebas de penetración y auditorías de seguridad; "Kali es una completa reconstrucción de BackTrack Linux desde la base hacia arriba, y se adhiere completamente a los estándares de desarrollo de Debian" [5].

El proceso de desarrollo la aplicación aportará beneficios para la Fundación Universitaria de San Gil UNISANGIL, debido a que contribuirá con la posibilidad de tener variedad de documentación sobre una aplicación construida en software libre, desarrollada en lenguajes como Java para crear una interfaz amigable y sencilla al usuario, y SQLite [6] para su base de datos. Se emplea un entorno de desarrollo Netbeans y un paradigma de programación MVC (modelo, vista, controlador). Esta investigación ayudará a los futuros estudiantes que se inclinen por estos temas y desarrollen innovadoras aplicaciones con base a esta.

Al inicializar el aplicativo Forense Soft muestra la ventana del login, que da la opción de insertar los datos de un usuario creado con anterioridad para iniciar sesión o la opción de nuevo usuario si no se encuentra registrado, como se presenta en la figura 1.

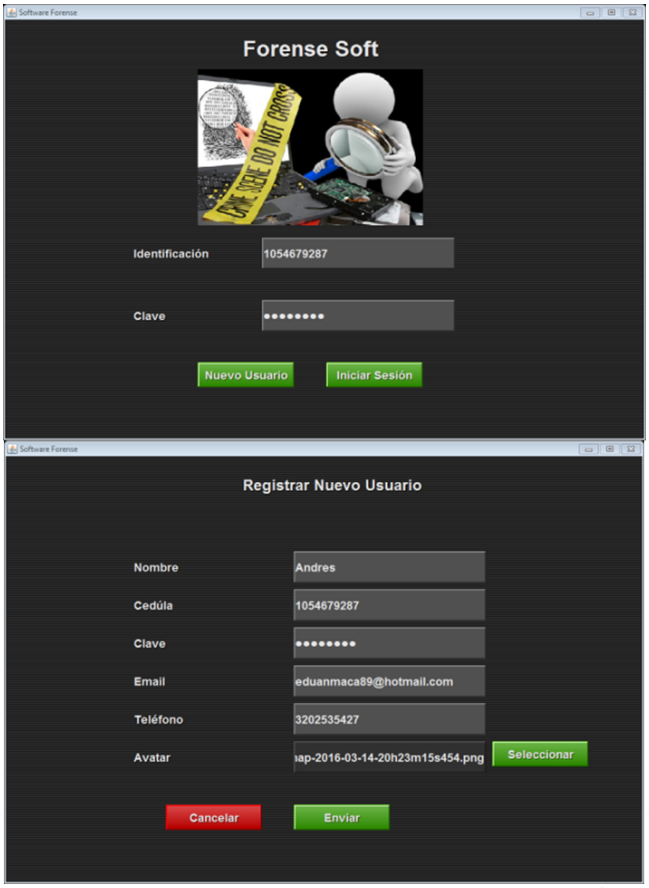

Fig. 1 Validacion y seguridad de usuario con el aplicativo Forense Soft.

Continuando el proceso, se procede a la carga de archivos en el aplicativo Forense Soft como se muestra en la figura 2.

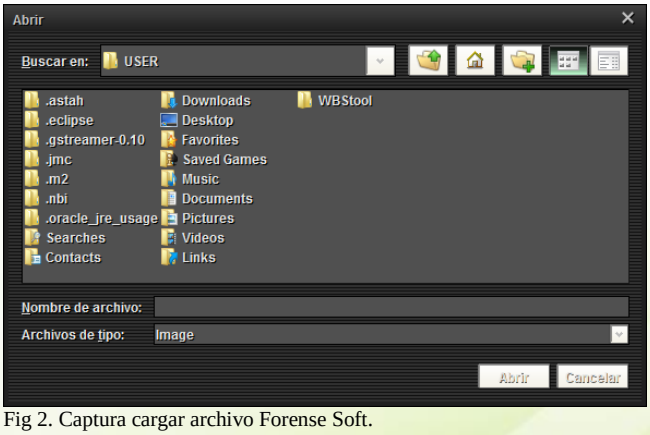

Fig 2. Captura cargar archivo Forense Soft.

Una vez cargados los archivos se procede a analizarlos, mediante el análisis de malware [7], el cual consiste en identificar los archivos maliciosos (figura 3). Así mismo, se realiza el análisis de metadatos que proporciona datos detallados para conocer el estado y condiciones en las que se encuentra el mismo junto con sus características principales (figura 4).

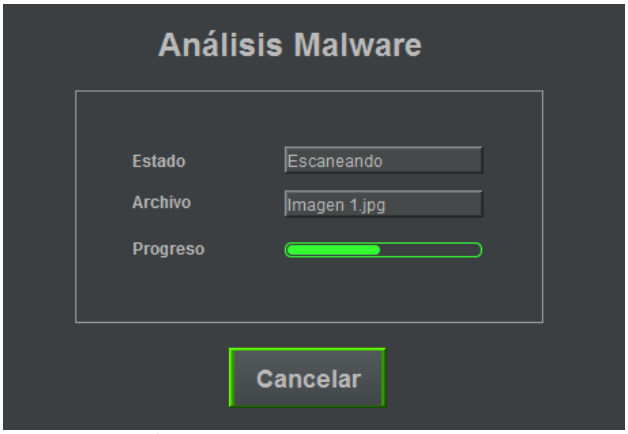

Fig 3. Captura análisis de malware.

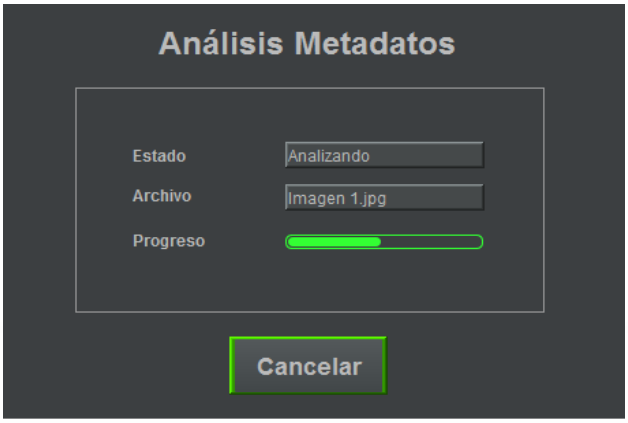

Fig 4. Captura analisis de metadatos.

Como producto de los análisis anteriormente mencionados se obtienen los reportes concretos de los archivos, para ser almacenados dentro de la base de datos de Fornse Soft y así poder llevar de forma ordenada y exacta el estado de cada uno de los archivos (ver figura 5).

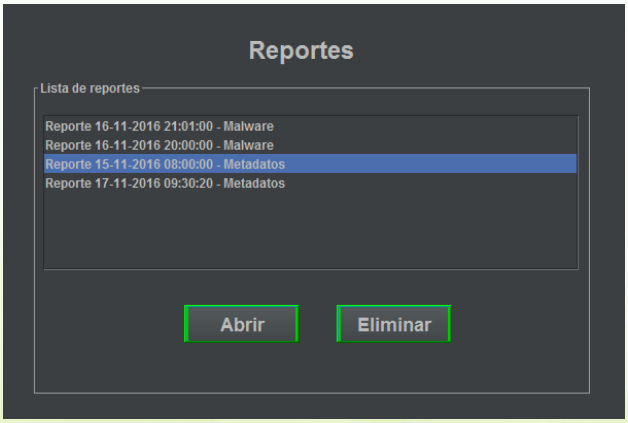

Fig 5. Captura de reportes Forense Soft.

III. METODOLOGÍA

Para el desarrollo de la aplicación se usó la metodología RUP: Proceso Unificado de Rational, una de las más utilizadas para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

Esta metodología, al estar basada en la evolución de prototipos ejecutables, se puede hacer un seguimiento de que cada paso por el ciclo de vida produce una versión del producto que incrementalmente se va refinando en las iteraciones de las diferentes fases. Si llegado el final del ciclo de vida de RUP, el producto no cumple con los objetivos planteados, se puede realizar un ciclo más para refinar, corregir y agregar funcionalidades que lleven al software a cumplir con las expectativas o cancelar el proyecto con base en los resultados obtenidos. Lo que indica que, con un enfoque iterativo e incremental, se tiene un mejor manejo de los riesgos y un refinamiento más efectivo del producto final.

En la siguiente figura 6, se muestra las etapas del proceso que se llevó a cabo para desarrollar el proyecto y una breve explicación de cada una.

La fase de planeación fue imprescindible para generar una idea global de las etapas y componentes del proyecto al comienzo del mismo, definiendo su alcance, exclusiones, lista de actividades y entregables; estableciendo una guía fundamental de buenas prácticas.

Posteriormente, se establecieron los requerimientos del sistema, en otras palabras, se elaboró una lista de las funcionalidades del aplicativo, creando una visión más clara de sus componentes.

En la etapa de arquitectura, teniendo los requerimientos, se procedió a diseñar los distintos modelos del sistema como componentes, paquetes, clases, modelo BD; con los cuales se dio mayor claridad a la estructura física que tendrían los archivos del aplicativo y su interacción.

En la fase de análisis y diseño, se realizaron los diagramas de secuencia, donde se estableció la interacción detallada de cada una de las partes del sistema, en otras palabras, el proceso que éste hace al momento que el usuario final selecciona alguna opción.

Posteriormente, en la fase de implementación fue donde se creó cada uno de los módulos pertinentes al proyecto, usando para ello tecnologías como Java y Netbeans. Es la etapa quizás más importante del proyecto ya que es la de mayor duración y de la cual depende el éxito del mismo.

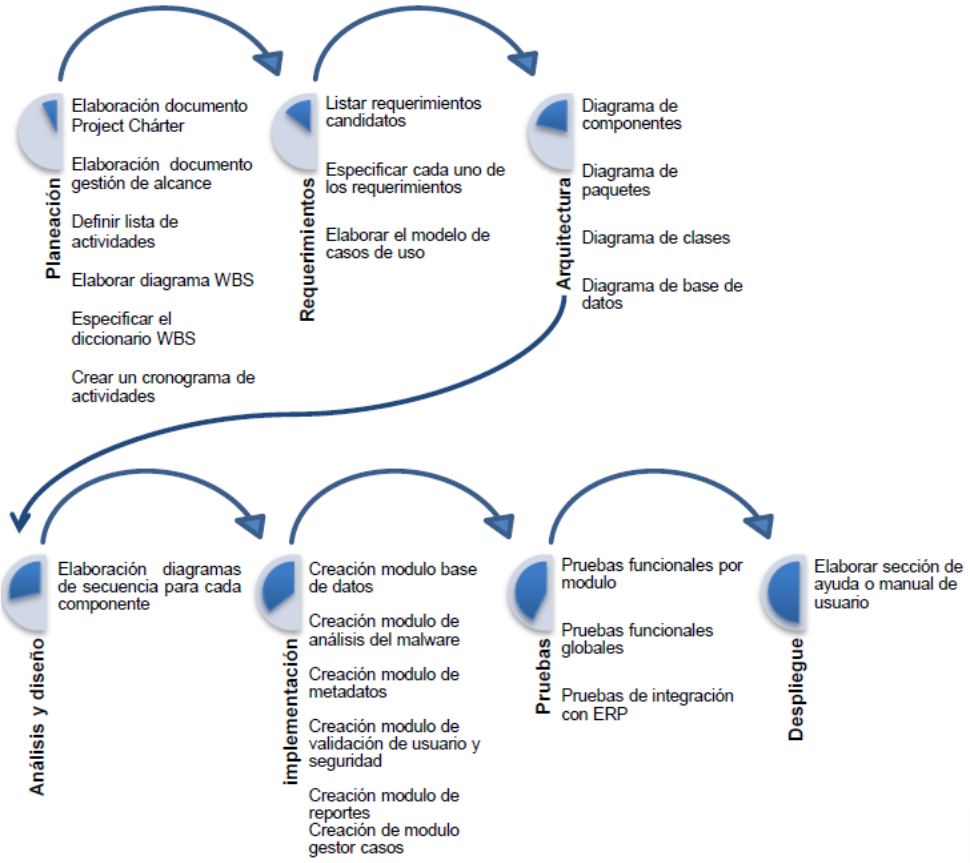

Fig. 6 Etapas y proceso de desarrollo del aplicativo.

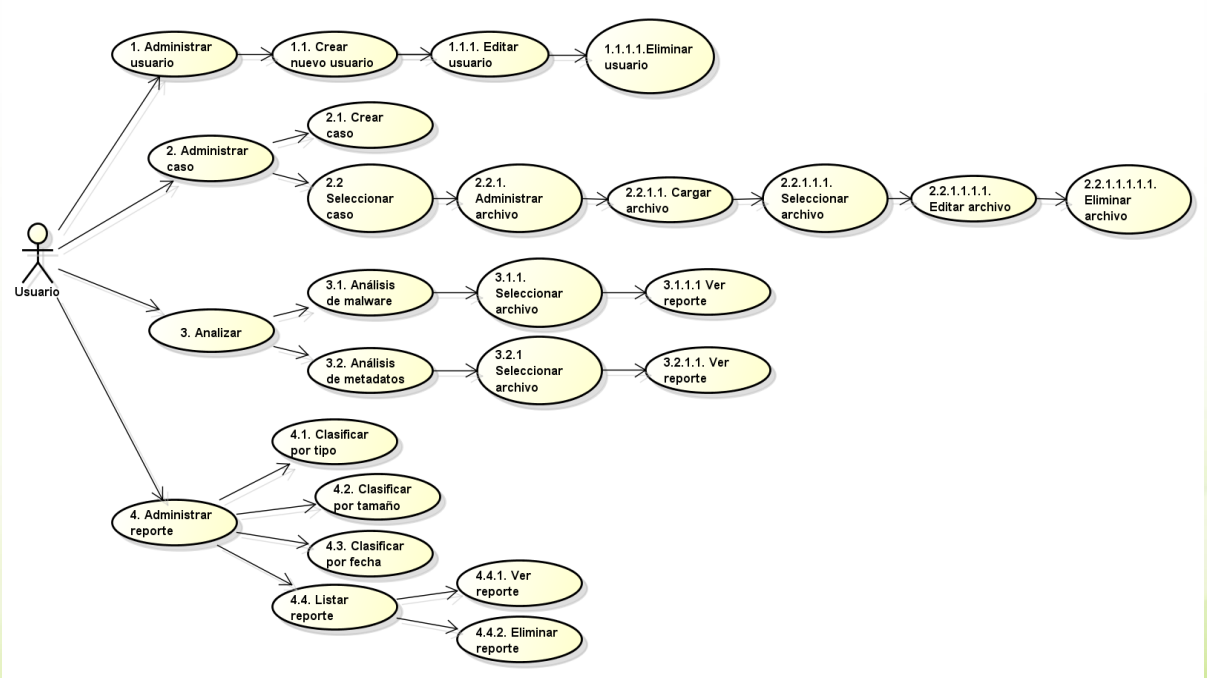

Fig. 7 Diagrama casos de uso.

Sin duda, las pruebas en todo desarrollo de software son importantes ya que aseguran la calidad del producto a entregar, en la fase de pruebas se encontraron distintos errores y se corrigieron, dejando así una aplicación totalmente funcional.

La fase de despliegue, fue donde se elaboró una sección de ayuda con cada una de las secciones explicadas para el usuario final.

IV. RESULTADOS

En cada una de las fases de desarrollo se obtuvieron resultados o entregables importantes que ayudan a especificar de mejor manera el software; a continuación se resumirán los más destacados

Se puede observar en la figura 7, el diagrama de casos de uso de la fase de requerimientos, donde el Usuario tiene a su disposición las funcionalidades del sistema como administrar de usuarios, casos y reportes al igual que analizar (malware y metadatos); cada uno de ellos con sus funciones determinadas y simples de acceder, creando así una aplicación con gran usabilidad, rapidez y confiabilidad.

En la fase de arquitectura, el diagrama de componentes de la figura 8 fue el primero en crearse, ya que muestra una visión general de la estructura del sistema pero, a la vez muy clara, de manera que permite definir lenguajes e interacciones entre servicios. Se puede observar como la interfaz realizada con Java Swing interactúa por medio Java controlador con Java Modelo, al que se le implementó toda la lógica del sistema, atributos y métodos para que a su vez interactúe con la base de datos local SQLite.

El diagrama de secuencia de la figura 9, del módulo Cargar Archivo, y que corresponde a la fase análisis y diseño fue de gran ayuda [9], ya que con sus interacciones básicas se tuvo claridad al momento de la implementación, en este caso, para seguir las pautas de adicionar y adjuntar elementos y así obtener los datos adecuadamente. De la misma manera se creó un diagrama de secuencia para cada uno de los módulos.

La implementación es la fase más ardua en el proyecto, aunque siguiendo las pautas de los anteriores diagramas se obtiene un bosquejo y reglas que indican por dónde empezar, la figura 10 muestra el menú principal de la aplicación donde se encuentran cada uno de los botones para acceder a sus respectivas funciones.

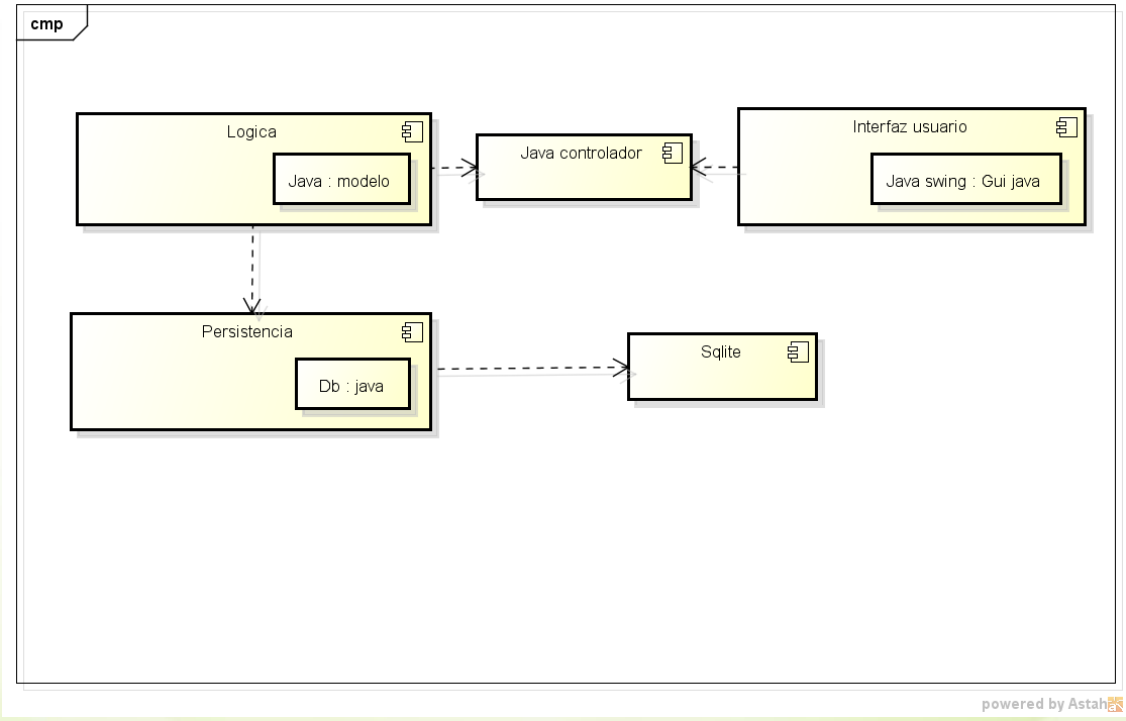

Fig. 8 Diagrama de componentes.

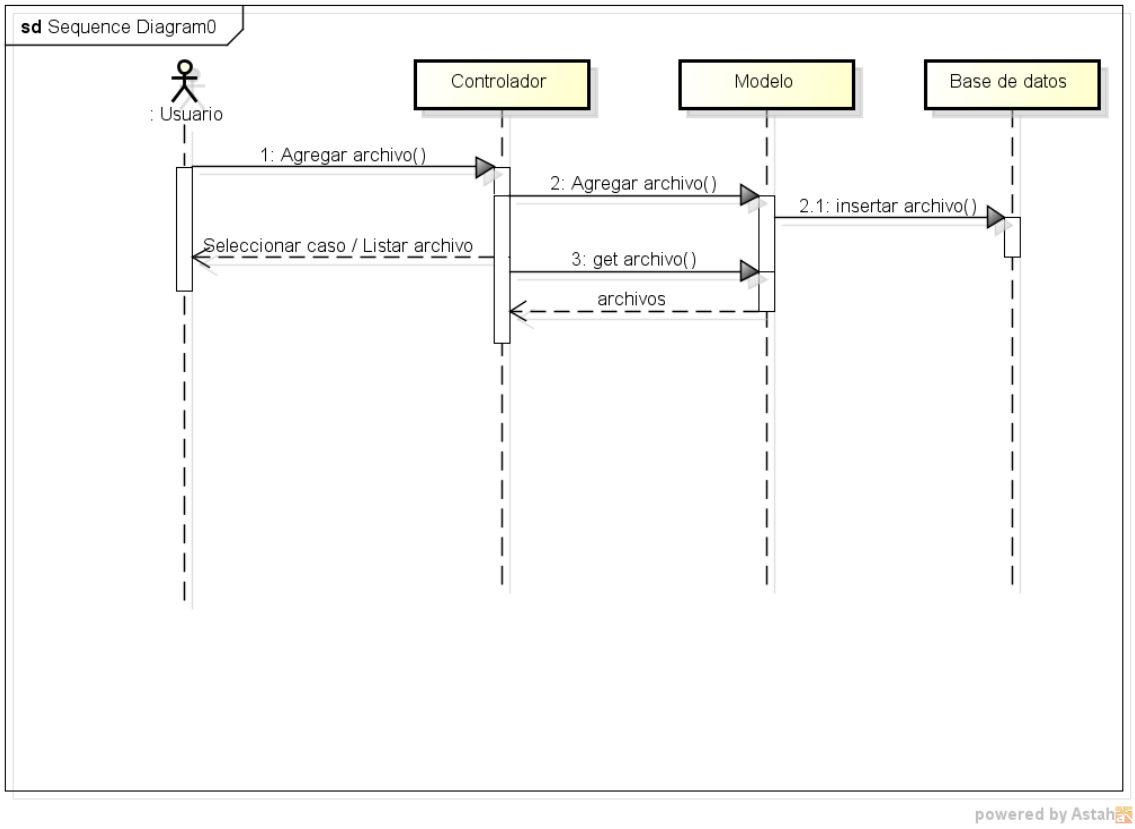

Figura 9. Diagrama de secuencia módulo Cargar Archivo

.

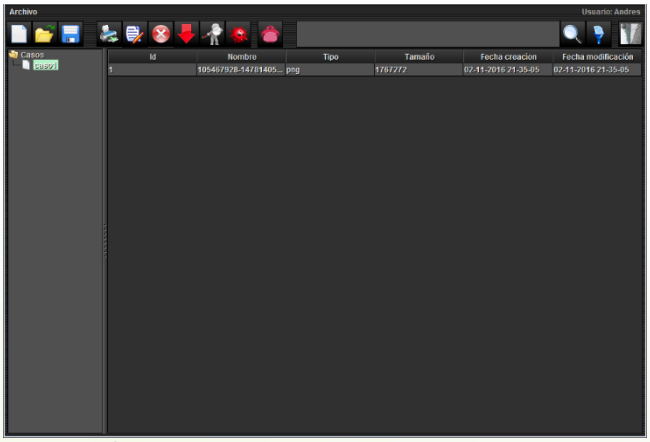

Fig. 10 Menú principal.

V. CONCLUSIONES

El uso de herramientas como Netbeans, Kali Linux, Sqlite, Autopsy y JavaScript son un soporte importante para la realización de proyectos a través de plataformas de software libre.

La plena identificación de los requerimientos del software generó la creación de límites de su funcionalidad, permitiendo la planeación de una serie de actividades y logrando así profundizar los temas para alcanzar con éxito el desarrollo de este proyecto.

La planeación y diseño de los diagramas, en la fase de planeación del proyecto, son la base fundamental del software, ya que permiten establecer un enfoque claro de lo que se va a desarrollar, consiguiendo minimizar desde el inicio posibles errores y fortaleciendo una base sólida para la fase de programación.

En la implementación se usaron plugins de terceros para brindar las funcionalidades del análisis de metadatos y del análisis de malware, siendo así la mejor opción puesto que ya contienen métodos implementados para tales fines, así como librerías y dependencias exclusivas de Windows. De igual manera, se realizaron métodos pertinentes para la generación de reportes ya que los plugins los generan de forma ambigua y en un formato no ideal para el usuario final.

Mediante la etapa de pruebas, en el desarrollo de software, se evidenciaron los errores e insuficiencias, efectuando la documentación necesaria y logrando de esta manera la corrección de errores de programación y brindando un producto final de calidad.

Con el desarrollo del proyecto se presentan las bases en cuanto a procesamiento de la información encontrada en una investigación forense; como ya es de conocimiento público, los atacantes y sus técnicas van progresando a pasos agigantados y siempre están al tanto de la tecnología; por esta razón, existe el reto permanente de investigar, profundizar y mejorar los diferentes procedimientos y prácticas que permitan realizar un mejor análisis de los archivos sospechosos y llevar una bitácora de los mismos, en donde se registren sus hallazgos, las hipótesis y posibles acciones a realizar.

A la hora de actualizar el proyecto a una versión más reciente de Kali Linux y Autopsy se debe tener cuidado en la reutilización del código, ya que cada nueva actualización presume cambios que pueden dejar como ineficientes o ineficaces el uso de algunas clases o instancias.

REFERENCIAS

[1] A. Estrada Canedo. "La informática forense y los delitos informáticos". *Revista Pensamiento Americano*. vol. 4, pp. 81-88, enero – junio 2010.

[2] A. E. Caballero Quezada. Hacking con Kali Linux. 2018. [Online]. Available: http://bit.ly/1C4vpo4.

[3] A. Montoya Rojas. "La informática forense como herramienta para la aplicación de la prueba electrónica". *Revista CES Derecho*, Universidad CES - Facultad de Derecho. ISSN 2145-7719. 2010. [Online]. Available: http://revistas.ces.edu.co/index.php/derecho/article/view/1289

[4] Ministerio del Interior. Ley 1273 de 2009. 2009. [Online]. Available: http://bit.ly/1tXP43h

[5] (2014) Cubieboard. Instalación de kali linux en raspberry pi y cubieboard. [Online]. Available: https://computadorasdeplacareducida.wordpress.com/category/cubieboard/

[6] SQLite, (s. f). About SQLite, [Artículo de internet], http://bit.ly/bsolld.

[7] ZAMUDIO, L, (2009), UML (Unified Modeling Language). [Online]. Available: http://bit.ly/17IgUGh.

[8] Análisis de malware. Welivesecurity. [Online]. Available: https://www.welivesecurity.com/la-es/category/analisis-malware/

[9] R. A. Castro Gil. Estructura básica del proceso unificado de desarrollo de software. *Sistemas & Telemática*. Universidad Icesi. p.p. 29-41. 2004. [Online]. Available: http://www.icesi.edu.co/contenido/pdfs/rcastro_estructura-bas-puds.pdf

[10] K. Cevallos. UML Diagrama de secuencia. 2015. [Online]. Available: https://ingsotfwarekarlacevallos.wordpress.com/2015/07/07/uml-diagrama-de-secuencia/