Instituto de Computación - Facultad de Ingeniería
Universidad de la República

Proyecto de Grado

Carrera Ingeniería en Computación

Análisis de evidencia digital obtenida de dispositivos móviles

Informe ejecutivo

Estudiantes:
Germán González
Horacio Pérez

Supervisores:
Gustavo Betarte
Marcelo Rodríguez

Diciembre 2015

Resumen

El análisis forense digital de dispositivos móviles es un área que ha venido creciendo considerablemente en el último tiempo debido al aumento en el uso de los dispositivos móviles como parte de la vida cotidiana de las personas y organizaciones. La particularidad de estos dispositivos conlleva varios desafíos nuevos en lo que respecta a esta disciplina. Diferenciándose del análisis forense digital tradicional en algunos procedimientos, como por ejemplo, para la extracción o para la preservación de la evidencia.

En este trabajo se realiza un estudio del estado del arte del análisis de la evidencia obtenida de dispositivos móviles donde se describen los tipos de evidencia que se encuentran en éstos, tipos de análisis que se pueden realizar (análisis temporal, relacional y funcional), herramientas disponibles, lenguajes y un modelo de referencia propuesto por Dorothy E. Denning [1]. Por otra parte, se plantean distintos tipos de metodologías a seguir para realizar los análisis.

Uno de los problemas detectados en el estudio realizado es el gran volumen de información a analizar en el transcurso de las investigaciones, por lo que se propone desarrollar una herramienta que asista al investigador en el proceso de análisis de la evidencia, reduciendo el volumen a manejar y brindando información hacia donde focalizar la atención al momento de iniciar la investigación, donde la metodología usada para la reducción fue aplicar una adaptación del modelo mencionado anteriormente.

Otros problemas observados fueron la falta de interoperabilidad entre las herramientas y la heterogeneidad de las representaciones de los datos en los dispositivos móviles, por lo que el desarrollo de la herramienta debió tener en cuenta estas problemáticas. Un requerimiento que se estableció fue que resultara fácilmente extensible tanto para la carga de cualquier fuente de datos como para la exportación de resultados de modo de lograr la interoperabilidad entre herramientas. La puesta en práctica de esto fue a través de la carga de los datos del caso de estudio con objetos generados a través de la herramienta *Android Inspector* [2] desarrollada por el grupo del proyecto de grado *"Recolección de evidencia digital sobre dispositivos móviles"*.

Se comprobó la utilidad de la herramienta con un caso de estudio donde se observa que se reducen notoriamente los eventos a analizar y, a su vez, donde se desprenden pistas del resultado para proseguir con el análisis de más evidencia en el transcurso de la investigación.

Por último, aquellas cuestiones que se discutieron y se consideraron de interés pero quedaron fuera del alcance de este trabajo, se mencionan como líneas a seguir en el trabajo futuro.

Índice

Índice de tablas

Lista de Figuras

1. Introducción

1.1. Motivación

Los smartphones y tablets han incorporado la complejidad de la computadora de uso personal, su penetración en el mercado ha crecido significativamente así como la dependencia de las personas sobre estos dispositivos, en particular los smartphones, dada las enormes ventajas que brindan. Concretamente en el mercado de la telefonía móvil en todo el mundo, en el segundo trimestre de 2015 [3] los vendedores entregaron 341,5 millones de smartphones, donde el sistema operativo preponderante fue Android, con el 82,8 % de las ventas. Esto tiene como consecuencia el efecto colateral de posicionar los smartphones con Android como blancos interesantes de ataque.

Según Kaspersky Lab [4], "Las piezas de malware para dispositivos móviles interesan mucho a los cibercriminales y el número de las aplicaciones maliciosas para smartphones está en constante crecimiento.". Además, señala que en el 2013 han sido detectados 143.211 programas maliciosos para dispositivos móviles, indicando que son 4 los principales intereses del cibercrimen y dando porcentajes concretos de 3 de ellos:

- **robar dinero (33,5 %):** SMS y llamadas a números Premium, intercepción de contraseñas de un solo uso (OPT, mTAN) utilizadas en servicios bancarios y de pago para móviles, robo a través de los servicios de pagos móviles (QIWI) y robo de bitcoins.
- **robar datos (20,6 %):** Robo de cuentas online presentes en los dispositivos, lectura de SMS y mails de forma remota, y robo de fotos y documentos.
- **ganar dinero (19,4 %):** Uso de los smartphones en una botnet, para que envíen SMS o mails de spam o para que ejecuten ataques DDOS.
- **espiar:** Localización, vigilancia de micrófonos y cámaras, y análisis de los mensajes y del registro de llamadas.

En el primer trimestre de 2014 [5], la cantidad de amenazas para Android superó el 99 % del total de amenazas para dispositivos móviles y se han descubierto:

- 1.258.436 paquetes de instalación maliciosos,
- 110.324 nuevos programas maliciosos para dispositivos móviles,
- 1.182 troyanos bancarios móviles.

Desde una óptica diametralmente opuesta, se pueden considerar a los smartphones como una herramienta para delinquir o simplemente un medio de comunicación entre individuos que se asocian para realizar algún ilícito. Es decir, utilizando alguno de los mecanismos que éstos proveen como ser las llamadas, SMS, e-mails, redes sociales, entre otros.

Lo expresado anteriormente evidencia una creciente tendencia de encontrar dispositivos (mayoritariamente smartphones) involucrados o al menos sospechados de participar en un incidente, ya sea como medio para cometer un ilícito o como víctima de un ataque, lo que dispara la necesidad de un análisis forense digital, presentando un abanico de desafíos que motivan el presente trabajo.

1.2. Objetivo

En líneas generales, el objetivo de este proyecto es hacer un estudio del estado del arte de la temática planteada, detectar los desafíos que ésta presenta y, en consecuencia, contribuir con una herramienta que facilite el trabajo del investigador forense en la etapa de análisis del proceso de análisis de la evidencia digital.

1.3. Metodología seguida

Este trabajo, enmarcado dentro de la línea de investigación que se viene desarrollando por el grupo de seguridad informática (GSI) de la Facultad de Ingeniería de la Universidad de la República en el área informática forense, se centra en el análisis de la evidencia digital obtenida de dispositivos móviles.
Además, complementa el trabajo referido a la extracción de datos de dispositivos móviles realizado por otros estudiantes de grado.

Con el fin de obtener una interiorización mayor en la temática forense, en paralelo con la realización del estudio del estado del arte, el grupo optó por acompañar el curso de posgrado *Metodologías para el Análisis Forense Informático* (MAFI) dictado por los tutores de este proyecto en el marco del CPAP, además de realizar un módulo taller enfocado en la generación de un repositorio para investigación y entrenamiento en análisis forense digital.

Para la confección del estudio del estado del arte se trató de considerar fuentes que brindaran cierto grado de confiabilidad como, por ejemplo, el National Institute of Standards and Technology (NIST) [6], Mitre [7] y Digital Forensics Research Workshop (DFRWS) [8]. Luego de tener una perspectiva del proceso de análisis forense digital y de los distintos desafíos encontrados en el contexto de la etapa de análisis, se definieron los objetivos que le darían dirección al resto del trabajo realizado, pasando por las etapas de desarrollo de software (análisis, diseño e implementación) de un prototipo en las que se aplicó un proceso iterativo e incremental.

Por último, se generó un caso de estudio con el fin de validar el prototipo realizado, poniendo a prueba su efectividad en un escenario lo más real posible.

1.4. Estructura del Documento

Con el objetivo de brindar una idea general de la organización y contenido del documento a continuación se presenta la estructura del mismo.

En la sección *Estado del Arte* se brinda un resumen del documento generado en la investigación del estado del arte. En ésta, se podrá encontrar la terminología, métodos de análisis, herramientas y desafíos que se deben sortear en el transcurso de una investigación forense focalizado en la etapa de análisis.

En la sección *Análisis* se define y analiza el problema a resolver. Se profundiza en aspectos específicos del análisis de la evidencia de dispositivos móviles usados como medio para realizar actividades ilícitas, se hace una abstracción de la representación de la evidencia a través de un modelo, se especifican los requerimientos y casos de uso que deberá satisfacer la herramienta a desarrollar, así como el alcance del trabajo realizado.

En la sección *Diseño* se especifican las decisiones tomadas para satisfacer los requerimientos planteados en la sección anterior.

En la sección *Implementación* se detallan los aspectos más relevantes asociados a las características de la plataforma y tecnologías de implementación elegidas, así como la descripción de cómo se hicieron algunas funcionalidades relevantes para el trabajo realizado.

En la sección *Caso de Estudio* se pretende validar la herramienta desarrollada para la cual se mostrará, usando un juego de datos relativamente representativo de la realidad, una simulación de un análisis llevado adelante por un investigador en el que éste la usará como un asistente para facilitar su tarea.

En la sección *Trabajo Futuro* se exponen funcionalidades interesantes que se le pueden agregar a la herramienta desarrollada que quedaron fuera del alcance del presente trabajo. Así como también se presenta una línea de análisis de cómo enfocar el estudio de malware en dispositivos móviles poniendo a estos como víctimas en un contexto de investigación forense.

En la sección *Conclusiones* se presentan las conclusiones, valga la redundancia, obtenidas al mirar en retrospectiva el trabajo realizado, partiendo desde el estudio del estado del arte, pasando por las dificultades encontradas, la evaluación y aportes de la herramienta desarrollada, hasta finalizar con las líneas de trabajo futuro que se proponen.

En la sección *Referencias* se podrá ver las distintas fuentes que se consultaron y que fueron fundamentales para la realización de este trabajo.

Por último, se encontrará la sección *Anexos*, en la que se pretende dar al lector interesado un mayor nivel de detalle de las características y funcionalidades que se implementaron en la herramienta desarrollada. Como, por ejemplo, la posibilidad de contar con un asistente para personalizar los parámetros de las operaciones de análisis que el plugin de detección de comportamiento sospechoso brinda.

2. Estado del Arte

2.1. Proceso de análisis forense digital

En esta sub-sección se presentan nociones generales del proceso de análisis forense digital con el fin de definir el marco en el que se desarrollará este trabajo.

2.1.1. Análisis forense digital

La Ciencia Forense proporciona procedimientos y métodos de análisis que permiten identificar, recuperar, reconstruir y analizar evidencias de lo ocurrido, aportando técnicas y principios necesarios para realizar una investigación frente a un determinado evento.

Específicamente, la Ciencia Forense en informática se puede encontrar bajo las denominaciones: Análisis Forense Digital, Análisis Forense Informático, Informática Forense, Computer Forensics, Digital Forensics o Cómputo Forense. Esta disciplina es relativamente nueva y se aplica tanto para la investigación de delitos "clásicos" como ser homicidios, fraude financiero, narcotráfico, entre otros; así como para la investigación de delitos meramente relacionados con las tecnologías de la información como ser la piratería de software, distribución de pornografía infantil, etcétera.

Si bien existen varias definiciones de Análisis Forense Digital, se usará la dada en [9], "Conjunto de principios y técnicas que comprende el proceso de adquisición, conservación, documentación, análisis y presentación de evidencias digitales y que llegado al caso puedan ser aceptadas legalmente en un proceso judicial".

2.1.2. Evidencia digital

Definición

Existen varias definiciones de evidencia digital, en este trabajo se dará la siguiente: "la evidencia digital comprende todos los datos digitales que permitan establecer que un crimen ha sido cometido o proporcionen un vínculo entre un crimen y su víctima, o un crimen y su autor" [10].

Propiedades de la Evidencia

La evidencia debe poseer distintas propiedades

- **Admisible**: Las pruebas deberán estar relacionadas con el hecho que se ha demostrado.
- **Auténtica**: Una evidencia digital será autentica siempre y cuando se cumplan dos elementos:
  1. Demostrar que dicha evidencia ha sido generada y registrada en el lugar de los hechos
  2. La evidencia digital debe mostrar que los medios originales no han sido modificados, es decir, que los registros corresponden efectivamente a la realidad y que son un fiel reflejo de la misma.
- **Completa**: Deberá demostrar acciones del atacante o su inocencia.
- **Confiable**: Debe ser clara y comprensible para los jueces o contraparte.
- **Creíble**: No pueden existir dudas sobre la veracidad o autenticidad de la evidencia.

Clasificación de la evidencia

Dada la diversidad de orígenes de la evidencia es necesario clasificarla, tomando como referencia el RFC 3227 [11] que establece que la recolección se debe realizar según el orden de volatilidad de la evidencia comenzando por las más volátiles hasta las menos volátiles, la misma se clasificará según su volatilidad.

Un ejemplo de orden de volatilidad de un sistema típico es el siguiente:

- Registros del CPU, cache.
- Tablas de ruteo, cache ARP, tabla de procesos, Memoria RAM.
- Sistemas de Archivos temporarios
- Discos
- Datos de monitoreo y logs remotos que sean relevante al sistema en cuestión
- Topología de la red, configuración física.
- Medios de archivos

2.1.3. Metodología de trabajo

El desarrollo de una investigación forense digital se divide en etapas o clases, las cuales pueden variar dependiendo del autor. En el paper "Analyses of the State-of-the-art Digital Forensic Investigation Process Models" [12] se presentan diferentes modelos de procesos de investigación forense digital así como también una comparación entre ellos. Si se desea consultar una versión resumida de estos se puede ver la tabla *Comparación de modelos de procesos de investigación forense digital* del documento de Estado del Arte [13].
En este trabajo se tomará el modelo definido en el reporte generado en el DFRWS, el que consta de siete etapas, a saber: Identificación, Preservación, Recolección, Examinación, Análisis, Presentación y Decisión [14].
En la figura 1 se ilustra el proceso lineal que dichas etapas establecen, resaltando la etapa de análisis que será el centro de este trabajo.

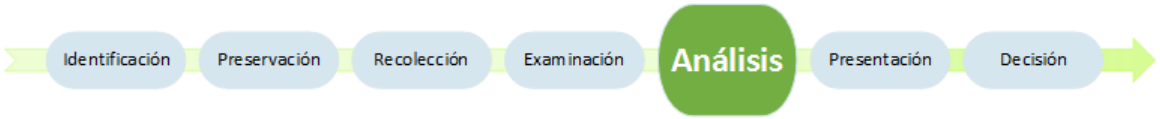

Figura 1: Proceso de análisis forense digital definido por DFRWS

Como el lector habrá notado, el uso del término análisis aparece tanto para referirse a la etapa como a todo el proceso, esto muchas veces da lugar a confusión, por lo que de ahora en más cuando se mencione el término análisis, salvo que se aclare explícitamente, será en referencia a la etapa y no a todo el proceso.

Otra puntualización que se hace es que la evidencia a analizar que se considerará de interés es la que corresponde al dispositivo como objeto usado para generar un incidente y no a la del dispositivo como víctima de un ataque.

2.2. Análisis forense de la evidencia digital

Para poder llevar a cabo el análisis de la evidencia digital es necesario aplicar un examen a los datos recolectados, éste tiene lugar en la etapa de examinación, obteniendo los resultados mediante la aplicación de métodos científicamente establecidos, los cuales deben describir el contenido y estado de los datos completamente, incluyendo la fuente y la importancia potencial de estos.
Luego, estos datos resultantes se deben pasar por una etapa de reducción, donde se separan aquellos que son relevantes de los irrelevantes para la investigación.
En estas condiciones se puede comenzar el análisis que será realizado por el investigador, que es la persona (pudiendo ser más de una) que participa directamente en la realización de la investigación. [15]

Antes de comenzar con el análisis propiamente dicho, se presentan algunas metodologías y enfoques de análisis que pueden seguirse.

Análisis forense dudoso

La evaluación de la evidencia disponible de manera objetiva e independiente de las interpretaciones de los demás, para determinar su verdadero significado se conoce como "Análisis forense dudoso [1]" [10]. El objetivo es identificar los errores u omisiones que puedan haberse realizado. Es fundamental revisar las pruebas de entrada lo más objetivamente posible, cuestionar todo y asumir nada. En muchas ocasiones, la evidencia será presentada a un investigador junto con una interpretación (por ejemplo, esta es la evidencia de una intrusión informática o amenaza de muerte). Antes de confiar en las pruebas reunidas por otros, es imprescindible evaluar su fiabilidad y su significado. Un beneficio adicional del análisis forense dudoso es que el investigador se familiarice con todo el acervo probatorio del caso.

Análisis Forense Diferencial [16]

El análisis forense diferencial es una práctica que viene en aumento, esto se debe al gran crecimiento de las aplicaciones web que hacen más difícil determinar lo que se encuentra presente en un lugar determinado de la red. En consecuencia, los investigadores se centran en que los medios han cambiado entre dos instantes de tiempo, para ello realizan comparaciones de imágenes forenses diferentes y reportan las diferencias encontradas entre ellas, logrando una reducción de la cantidad de información a analizar. Algunos casos en donde se suele utilizar el análisis forense diferencial son aquellos en los que se aplica ingeniería reversa, por ejemplo, cuando se trata de inferir el comportamiento de un malware. Otro ejemplo es cuando un administrador de la red compara mes a mes el resumen del tráfico para aprender como evoluciona la demanda de ancho de banda.

Análisis automatizado

Dado que el análisis de la evidencia es algo que requiere del razonamiento humano, es un proceso que, en sus orígenes, consumía mucho tiempo. Con el pasar de los años se han ido construyendo herramientas para automatizar parte del análisis. Cabe destacar que este tipo de herramientas de automatización no pretenden ser un sustituto para el análisis forense completo realizado por un investigador calificado y con experiencia, sino lo que se pretende es ayudar con la reducción de los tiempos invertidos en dicha etapa. Un ejemplo de ello puede ser el trabajo *"An automated timeline reconstruction approach for digital forensic investigations"* realizado por Christopher Hargreaves y Jonathan Patterson [17]. Ellos observan que la extracción de las marcas de tiempos de una imagen de disco para generar una línea de tiempo puede estar compuesta de varios millones de eventos de "bajo nivel", por ende, proponen una técnica para reconstruir automáticamente los eventos de "alto nivel" (por ejemplo, la conexión de un dispositivo USB) a partir de una serie de eventos de "bajo nivel". Además, los eventos de "alto nivel" se pueden visualizar con herramientas existentes.

Enfoque del análisis

En el proceso de identificar los datos relevantes para la investigación, el análisis puede enfocarse en distintos tipos de evidencia según el contexto en el que participó el objeto investigado; entendiendo como contexto al vínculo entre el objeto y el suceso ocurrido, el objeto puede haber sido una víctima o puede haber sido utilizado como medio para realizarlo.
Dependiendo de estos contextos los tipos de evidencia relevantes para la investigación varían pudiendo así descartar algunos tipos con el fin de reducir el volumen de información a procesar.

[1] Dudoso refiere a todo lo que puede ser interpretado de más de una forma o cuando la interpretación es discutible. Un análisis forense dudoso es aquel en el que las conclusiones relativas a las pruebas físicas y digitales aún están abiertas a la interpretación.

Por ejemplo, se analizan los correos electrónicos intercambiados entre la víctima y el atacante en la búsqueda de un mensaje de amenaza. O se busca en la computadora de la víctima un malware que comprometió información privada de la misma.

Entrando en el análisis, se tiene que las piezas individuales de datos digitales pueden no ser útiles por sí mismas, pero cuando se combinan pueden surgir patrones.
Por ejemplo, si una víctima comprueba el correo electrónico en un momento específico o frecuenta un área en particular en Internet, una interrupción en este patrón podría ser un indicio de un acontecimiento inusual. O también, un delincuente sólo podría atacar los fines de semana, en un lugar determinado, o de una manera única.
Con esto en mente, para tener una visión más clara a la hora del análisis de las evidencias y detectar agujeros o discrepancias, se puede desarrollar una línea de tiempo relacionando diferentes eventos, un ejemplo se ilustra en la figura 2.

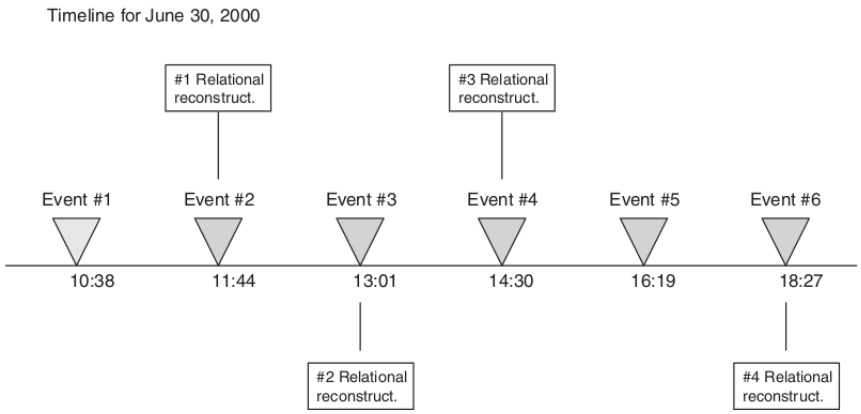

Figura 2: Vista conceptual de línea de tiempo y reconstrucciones relacionales [10]

La evidencia se utiliza para secuenciar eventos, determinar los lugares y caminos, establecer la dirección o el tiempo y/o la duración de la acción. Algunas de las pistas que se utilizan en estas determinaciones son temporales, cosas basadas en el paso del tiempo; relacionales, es decir, donde un objeto es en relación a otros objetos o con el delito; o funcionales, la forma en que algo funciona o cómo se ha utilizado. Esto da lugar a las categorías de análisis temporal, relacional y funcional, las que se describen a continuación.

- **Temporal (cuándo)**: Ayuda a identificar las secuencias y patrones en el tiempo en que ocurren los acontecimientos;
- **Relacional (quién, qué, dónde)**: Componentes del delito, sus posiciones y las interacciones;
- **Funcional (cómo)**: ¿Qué fue posible y qué no?

Análisis temporal [10]

Una pregunta natural que surge cuando se trata de sistemas informáticos, en el contexto temporal, refiere al grado de precisión con el que se registra el momento en el que tiene lugar un evento. Dado que las computadoras pueden representar el tiempo dentro de unos pocos milisegundos, el tiempo registrado se puede considerar preciso, lo que no quiere decir que éste se corresponda necesariamente con el momento en que el evento ocurre (por ejemplo, si el reloj de la computadora está mal configurado).

Pueden existir casos donde sea importante distinguir que un evento ocurrió primero dentro del mismo segundo, si bien en la mayoría la diferencia de segundos no son importantes e incluso puede ser suficiente un grado de precisión con un error de unos pocos minutos.

Enfoques para el análisis de la información temporal y la identificación de patrones

Lista Cronológica: La creación de una lista cronológica de eventos puede ayudar a aumentar la visión (identificando patrones y anomalías, arrojando luz sobre un crimen y conduciendo a otras fuentes de evidencia), de un investigador sobre lo ocurrido y las personas involucradas en un delito. Por ejemplo, un archivo de log con una gran brecha o entradas que están fuera de secuencia pueden ser una indicación de que el log fue manipulado.

Date	Activity
Day 1	Bondage/sadomasochistic (BDSM) Web sites viewed, probably by missing individual
Day 2	Hotmail e-mail correspondences of a sexual/BDSM nature with unknown individual
	IP address indicates Virginia. At around the same time as Hotmail is checked; Web pages from BDSM sites visited
Day 3	Logs of online chat sessions show conversation of a sexual/BDSM nature with unknown individual; IP address indicates Virginia
Day 4	Driving directions obtained from Mapquest, address of destination in Virginia
	Files deleted
	No activity after 8 P.M.

Figura 3: Cronología de las actividades en el equipo de la víctima mostrando correspondencia de correo electrónico, sesiones de chat en línea, archivos borrados, búsquedas web de mapas y planes de viaje en línea [10]

Histograma de ocurrencia: La creación de un histograma de ocurrencia de eventos puede revelar un período de alta actividad que merece una inspección más cercana. La organización de las ocurrencias en una gráfica con días en el eje horizontal y la cantidad de ocurrencias en el eje vertical puede resaltar patrones repetidos y desviaciones de los eventos regulares. En la figura 4 se muestra un ejemplo de esto.

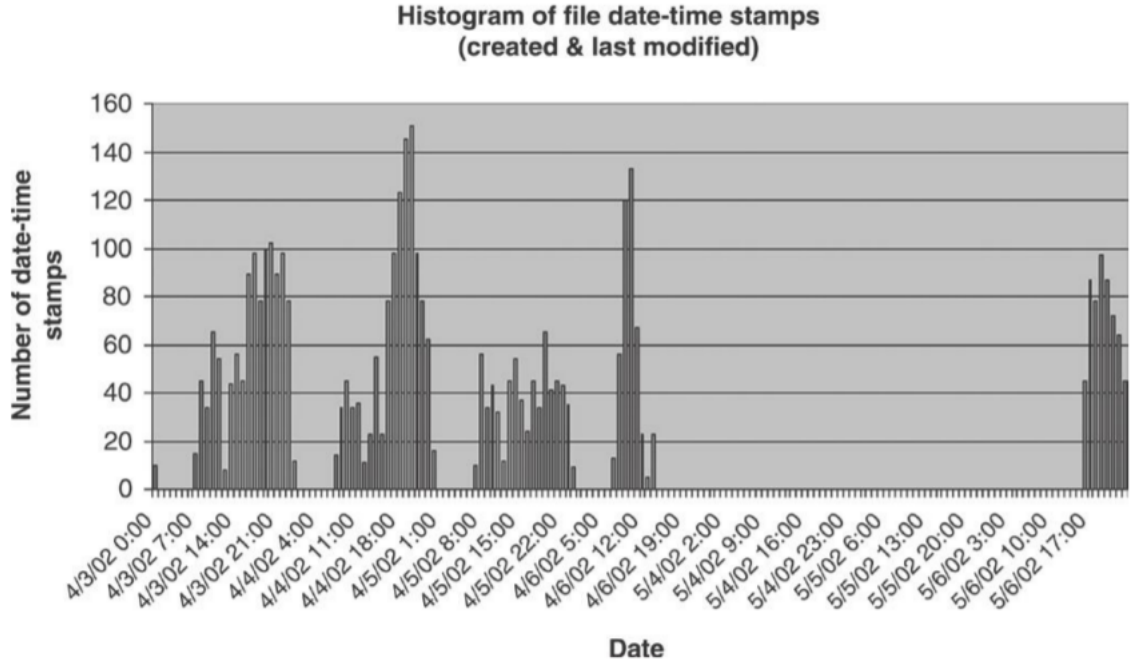

Figura 4: Histograma de date-time stamps (creación y última modificación) mostrando huecos durante los turnos sospechosos [10]

Cuadricula de patrones: El objetivo es acentuar los patrones en que se produjeron los acontecimientos. A modo de ejemplo, se muestra una cuadrícula con los e-mails enviados por el jefe de un grupo criminal a otros miembros del grupo en el lapso de algunos meses. Donde la comunicación sobre un plan criminal comenzó a mediados de junio, se interrumpió a principios de julio, y luego volvió a tomar impulso hasta la fecha límite del 11 de setiembre.

Email Address	Sun, Jun 16	Fri, Jun 21	Sun, Jun 23	Wed, June 26	Sat, Jun 29	Sun, Jun 30	Thu, Jul 11	Fri, Jul 26	Mon, Jul 29	Fri, Aug 2	Wed, Aug 14	Thu, Aug 15	Thu, Aug 29	Sun, Sep 8	Wed, Sep 11
Member1	xx				x	x							xxx	xx	x
Member2	xx		x	x			x		x		x	x	x	x	x
Member3	xx	x	x	x			x	x		xxx			x	x	x

Figura 5: Cuadrícula que muestra los mensajes de correo electrónico enviados por un sospechoso durante varios meses a varios miembros de un grupo criminal [10]

Otras formas de representación: Los investigadores deben buscar formas de representación visual de la información temporal tratando de reconocer patrones. Representar el tiempo en círculos concéntricos o en una espiral pueden generar patrones que se destaquen. Para ilustrar con un ejemplo se muestra la figura 6.

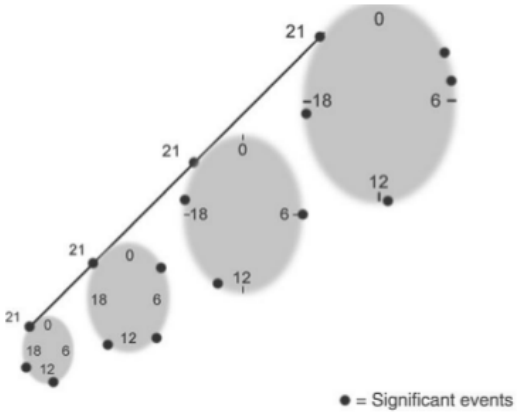

Figura 6: Imagen conceptual de relojes de 24 horas con MAC-time (Modification-Access-Creation) durante varios días, con una línea que une los eventos importantes en días secuenciales [10]

Análisis relacional [10]

El objetivo es identificar las relaciones entre los sospechosos, las víctimas y la escena del crimen. Esto es sumamente útil en la investigación de delitos que involucran redes de computadoras. A modo de ejemplo, en los grandes casos de fraude informático miles de personas y computadoras pueden estar involucrados, por lo que seguir la pista de las muchas relaciones entre los objetos es difícil, sin embargo, la creación de un diagrama relacional que represente las asociaciones entre las personas y las computadoras puede aclarar lo que ha ocurrido. Lo mismo se aplica a grandes registros de llamadas telefónicas o registros de tráfico de red, la creación de un diagrama de conexiones puede revelar patrones que permitan conocer el crimen.

Además, un diagrama relacional puede ser útil para localizar potenciales lugares de donde recolectar evidencia que anteriormente se pasó por alto. A modo de ejemplo, si un atacante informático obtuvo acceso no autorizado a una computadora dentro de una organización protegida del exterior con un firewall y luego ingresó al sistema de contabilidad, sin embargo, para obtener acceso al sistema de contabilidad

el atacante necesitó conocer una contraseña a la que solo tenía acceso un grupo reducido de empleados. En la figura 7 se muestra el diagrama relacional correspondiente.

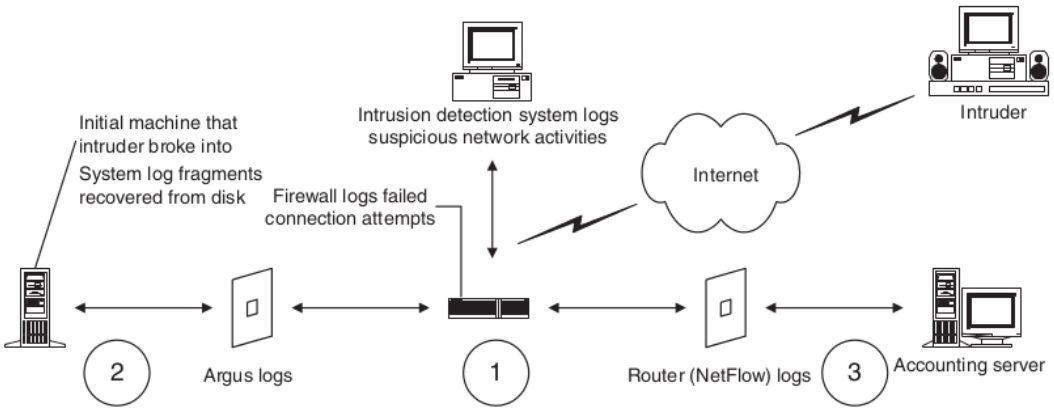

Figura 7: Diagrama que representa el intruso que accede al servidor de contabilidad [10]

Como se aprecia en el diagrama, potenciales fuentes donde recolectar evidencia digital son: el firewall, el IDS y los logs del router. Los registros del IDS y el firewall pueden mostrar la actividad del atacante en búsqueda de vulnerabilidades (a través del escaneo de la red), también permite ver si el atacante se concentra en algún sistema de la red en particular, así como los registros de tráfico entre la computadora comprometida y el servidor de contabilidad.

Un aspecto a tener en cuenta cuando se dispone de mucha información es que pueden relacionarse objetos o eventos que no están necesariamente vinculados, por ejemplo, un atacante compartió la escuela con el hermano de la víctima, pero esto puede ser solo una coincidencia. Está en manos de los investigadores decidir cuánto peso se le da a cualquier relación que encuentren. La creación de una reconstrucción relacional funciona mejor cuando se trata de un número reducido de entidades; a medida que el número de entidades y vínculos aumenta, se hace cada vez más difícil discernir conexiones importantes.

Análisis funcional [10]

El propósito de la reconstrucción funcional es considerar todas las explicaciones posibles para un determinado conjunto de circunstancias.

Para obtener una mejor comprensión de un delito o una pieza de evidencia digital, puede ser necesario determinar cómo se ha configurado un programa o equipo, esto puede arrojar luz sobre las pruebas digitales disponibles y puede ayudar a los investigadores a evaluar la fiabilidad y el significado de la evidencia digital. Por ejemplo, si un examen de una computadora muestra que la hora del sistema se desvía significativamente, perdiendo 2 minutos cada hora, esto debe ser tenido en cuenta en el desarrollo de la reconstrucción temporal de un caso. Si el equipo se ha reconfigurado después del crimen o un archivo de configuración del software no está disponible, no se podrá realizar un examen directo. Sin embargo, podría ser posible hacer una hipótesis basada en la evidencia asociada. Por ejemplo, si un archivo de registro muestra que el cliente de correo electrónico comprueba nuevos mensajes precisamente cada 15 minutos durante todo un día, una hipótesis es que fue automatizado en vez de manual.

2.3. Análisis forense de la evidencia digital en dispositivos móviles

Teniendo como motivador lo expuesto en el capítulo de introducción sobre el posicionamiento de los smartphones como blancos interesantes de ataque y la tendencia creciente de encontrarlos involucrados o al menos sospechados de participar en un incidente, se decide focalizar el análisis forense digital en los dispositivos móviles.

2.3.1. Tipos de evidencia en dispositivos móviles [10]

Los datos asociados a los teléfonos móviles se encuentran en varios lugares, memoria integrada, memoria extraíble y la tarjeta SIM (Subscriber Identity Module). No todos estos componentes estarán disponibles o serán necesarios para todas las investigaciones, pero en algunos casos pueden haber varias tarjetas SIM, varios medios extraíbles, o incluso más de un dispositivo móvil.

Dada la amplia gama de funcionalidades potenciales, cuando se trata de un dispositivo móvil en particular, es aconsejable determinar su conjunto de funcionalidades para obtener una mejor idea de qué tipo de evidencia digital se pueda conseguir.

Como mínimo, se puede esperar que los teléfonos móviles contengan las libretas de direcciones, registros de llamadas y mensajes de texto (SMS). Los mensajes de texto tienen la ventaja de proporcionar transcripciones completas, a diferencia de los registros de llamadas. Las marcas de tiempo (fecha y hora) de los SMS recibidos son generalmente confiables ya que son insertados por el proveedor del servicio de red en lugar de por el propio dispositivo móvil. Sin embargo, existen algunas desventajas en la investigación de los SMS, por ejemplo, no hay registro de que los mensajes hayan sido leídos, solamente es posible saber si han tenido acceso; además, los mensajes pueden estar incompletos si han sido borrados del teléfono.

A continuación se describirán los tipos de evidencia en dispositivos móviles y su aporte para la fase de análisis.

Libreta de direcciones

Ésta contiene la información de los distintos contactos, la misma sirve como apoyo para un investigador dando un panorama de la red de contactos de un sospechoso. La utilidad de la libreta de direcciones en el análisis se establece en la posibilidad de dar cuenta del vínculo que puedan tener el sospechoso y la víctima.

Historial de llamadas

La importancia del historial de llamadas es que brinda la posibilidad de ver con quién el sospechoso se ha comunicado, ya sea si llamó o lo llamaron y, a su vez, la duración de dichas llamadas. A la hora de analizar puede ayudar a realizar conclusiones indirectas.

Mensajes de texto y correos electrónicos

A diferencia de la libreta de direcciones y el historial de llamadas, los mensajes de texto y los correos electrónicos brindan información directa. Esto de debe a que se puede encontrar texto escrito o recibido por el sospechoso y pueden servir como pruebas en un juicio.

Calendario

En el calendario se puede observar las actividades realizadas y las planeadas a futuro por el sospechoso. Al momento del análisis se puede utilizar para vincular al sospechoso con lugares, fechas, tiempos y posibles testigos, cómplices o víctima.

Datos multimedia

Las fotografías, el audio o un vídeo pueden proporcionar las más convincentes evidencias digitales en un caso. Muchas veces los criminales filman su delito, en otros casos los que registran el acto son los cómplices. Sabidas estas conductas el investigador puede conectar al sospechoso/cómplice con la víctima. Pero no sólo en su contenido pueden contener elementos de prueba, sino también en la metadata de los archivos. Los datos EXIF (Exchangeable Image File Format) se pueden utilizar para determinar la fecha y hora del dispositivo en que fue sacada una foto y en algunos casos además se puede obtener la ubicación, de esta forma el propietario puede estar vinculado a la escena del crimen o a posibles coartadas.

Información de ubicación

Algunos dispositivos almacenan la ubicación de las torres celulares a las cuales se han conectado, un ejemplo de ello es el iPhone que almacena la ubicación de las torres celulares recientemente utilizadas en un archivo plist, donde almacena tanto la latitud y longitud de la misma como el timestamp de la conexión.
En aquellos dispositivos que cuentan con GPS se pueden encontrar rastros de las ubicaciones pasadas.
La utilidad de contar con las ubicaciones de un móvil en determinados momentos, es que sirve para relacionar temporal y espacialmente al sospechoso con ciertos lugares.

Códigos maliciosos

Ejemplos de este tipo de evidencia son las aplicaciones capaces de robar información bancaria y enviarla a terceros, otros tipos de malware más sofisticados interceptan los mensajes de textos asociados a las transacciones bancarias online y permiten robar dinero directamente de la cuenta de la víctima.
Encontrar rastros de un malware con estas características en un dispositivo puede servir, por ejemplo, para detectar un fraude.

Existen también aplicaciones capaces de enviar a terceros información acerca de los mensajes de textos, registros de llamadas, posiciones del GPS y el histórico de navegación por Internet.
En caso de existir evidencia de malware de estas características el investigador puede estar frente a un acto de espionaje y/o robo de información y podría direccionar su investigación al o los destinos a donde se envía la información.

Este tipo de dispositivos también son usados para lanzar ataques a otros sistemas, por lo tanto, la presencia de algunas aplicaciones como escaners de puertos pueden ser un indicio de que el mismo pudo haber sido utilizado en un ataque.

Evidencia de dispositivos móviles que se encuentra fuera de los mismos

No toda la evidencia que puede ser creada con un dispositivo móvil necesariamente se puede encontrar en el mismo. Existen casos donde los dispositivos fueron sincronizados con una computadora, en estos casos es posible encontrar información en la computadora donde se sincronizó.
También es posible solicitar información a los proveedores de servicio, éstos son capaces de proveer la información de conexiones con una torre celular o un punto de acceso WiFi. Asi, los proveedores de redes celulares tienen la posibilidad de reportar a los investigadores un resumen de los registros de llamadas, mensajes y transferencia de datos asociados a un dispositivo móvil.

Resumen
La siguiente tabla es un resumen de la evidencia potencial que se puede obtener de un dispositivo móvil.

Dispositivo	Ubicación o quién lo crea	Tipo de evidencia
Teléfono básico	Hardware	Fecha y hora del teléfono, IMEI [2] [3]
	Información creada por el usuario	Libreta de direcciones, SMS, Calendario
	Información creada por el celular	Registro de llamadas (recibidas, enviadas, perdidas)
Smartphone	Información creada por el usuario	Fotos (incluyendo datos EXIF), audio/video, mapas, posiciones GPS, mensajes de voz guardados, archivos almacenados en el sistema
	Información relacionada con Internet	Cuentas online, email, uso de Internet, Información de las redes sociales
	Aplicaciones de terceros instaladas	Sistemas de mensajería y comunicaciones de terceros, malwares,
Computadora	Información transferida	Datos respaldados, Aplicaciones de terceros, cuentas almacenadas
Proveedor de servicio telefónico	Información de seguimiento	Las torres donde se conectó, ubicaciones en todo momento
	Información de uso	Información de facturación, registros de llamadas, datos y usos de Internet, mensajes no repartidos,
Tarjeta SIM	Identificadores	IMSI, Identificador de tarjeta SIM (ICC-ID)
	Información de uso	SMS, discado abreviado, últimos números discados

Tabla 1: Evidencia potencial relacionada con dispositivos móviles [10]

2.4. Herramientas para forensia digital

Las herramientas forenses digitales ayudan a hacer el trabajo mucho más eficiente o incluso posible. Hay herramientas:

- para fines específicos así como también las que proporcionan un conjunto de funcionalidad más amplia
- pueden venir en forma de hardware, software o una combinación de ambos
- pueden ser comerciales que deben comprarse o pueden ser de código abierto que son de libre acceso
- tienen ventajas y desventajas

El uso de múltiples herramientas es también una manera de validar los hallazgos. Los mismos resultados, con dos herramientas diferentes, aumentan significativamente la fiabilidad de las pruebas.

2.4.1. Toolkits forenses

De todas las herramientas que se estudiaron, a continuación se describirán brevemente aquellas que cuentan con características que son relevantes para el objetivo de este proyecto dejando de lado las que no se alinean con el mismo. Por más detalles del resto de las herramientas se puede consultar el documento *Estado del Arte* [13].

`UFED Physical Analyzer` [18]
Es una aplicación desarrollada por Cellebrite Mobile Synchronization LTD y, según sus fabricantes, es la aplicación más avanzada de análisis, decodificación y generación de informes en la industria del análisis forense para dispositivos móviles. Esta aplicación incluye detección de programas maliciosos, funciones de decodificación y generación de informes, gráfico de línea de tiempo y capacidades de exportación de datos entre otras.

[2] IMEI: International Mobile Equipment Identity, es un código USSD pre-grabado en los teléfonos móviles GSM. Este código identifica al aparato unívocamente a nivel mundial, y es transmitido por el aparato a la red al conectarse a ésta.
[3] USSD: Unstructured Supplementary Service Data, es un servicio para el envío de datos a través de móviles GSM, al igual que el SMS.

Referidas al análisis concretamente alguna de las funcionalidades que presenta son:

- Proyecto Analítico: Permite ver estadísticas sobre las comunicaciones y permite la identificación de los puntos con fuerte relación
- Línea de tiempo: Permite visualizar eventos en el tiempo, ver las distancias entre los acontecimientos y ver el número de eventos dentro de un lapso de tiempo definido
- Búsqueda avanzada: Permite la búsqueda de información basada en texto libre o en varios parámetros
- Búsqueda por todos los campos de proyectos: Permite hacer una búsqueda rápida dentro de los datos decodificados
- Visor de conversación: Permite ver las comunicaciones entre entidades ordenadas por fecha y hora

En el sitio web de la empresa no hacen referencia a la posibilidad de compartir o no los resultados de los análisis llevados a cabo con otras herramientas que no sean del mismo fabricante.

`Autopsy` [19]
Es una interfaz gráfica para Sleuth kit [20] y otras herramientas forenses open source. La misma es de uso libre [4] y permite analizar tanto discos duros como imágenes de celulares, su arquitectura está basada en plugins, permitiendo así, la extensión de la misma a través del agregado de plugins de terceros o creando los propios tanto en Java como en Python.

Contiene un módulo para visualizar líneas de tiempo de objetos tales como mensajes, rutas de GPS, e-mails y llamadas, los cuales son cargados al sistema a través del módulo *"Android Analyzer Module"*. El mismo examina imágenes de celulares en búsqueda de bases de datos SQLite y otros archivos de dispositivos Android.

Con respecto al intercambio de información con otras herramientas, Autopsy permite la exportación en formato STIX [22]
Algunos objetos CybOX [23] que soporta son *"Email Message Object"*, *"File Object"* y *"URL History Object"*. Una limitante que tiene Autopsy es que no soporta todo el conjunto de objetos CybOX, y en lo que respecta a los dispositivos móviles se nota la ausencia de soporte para SMSs, agenda de contactos y llamadas.

`Oxigen Forensic Suite` [24]
Oxigen Forensic Suite es un kit de hardware y software forense móvil que posibilita la extracción y el análisis de datos de teléfonos celulares, smartphones y tablets.
Algunas características del software de análisis se detallan a continuación:

- Aggregated Contacts: Ofrece la posibilidad de analizar los contactos desde múltiples fuentes, como la agenda telefónica, mensajes, registro de eventos, Skype, chats y aplicaciones de mensajería.
- Links and Stats: Provee una herramienta para explorar conexiones sociales entre el usuario del dispositivo y sus contactos a través del análisis del registro de llamadas, mensajes SMS/MMS, correo electrónico y Skype.
- Timeline: Tiene la capacidad de organizar todas las llamadas, mensajes, eventos del calendario, datos geográficos y otras actividades en forma cronológica. Esta característica está disponible para un dispositivo o para todos los dispositivos que pertenecen a un mismo caso, revelando la lista completa de los eventos ocurridos en un caso.

En el caso de esta herramienta la única forma de compartir los datos para ser consumida por otra de terceros es a través de la exportación de los mismos en formato XML pero no se detalla si utiliza algún lenguaje estándar para expresarlos.

[4] Bajo licenciamiento Apache2 [21]

Resumen

Como resumen de las herramientas estudiadas se observa que existen muchas con una amplia variedad de funcionalidades para el análisis de la evidencia obtenida de los dispositivos móviles. Se puede ver que, la gran mayoría, son herramientas pagas y manejan datos en formatos privativos con lo cual se dificulta el intercambio de información entre herramientas de distintos fabricantes.
A su vez, se ven pocas herramientas gratuitas, de las que se encontraron se notó falta de soporte para la evidencia generada en dispositivos móviles e interfaces gráficas pobres.

2.5. Insumos para las herramientas forenses

2.5.1. Lenguajes para forensia digital

STIX

Structured Threat Information Expression (STIX) es un lenguaje para describir información de amenazas informáticas que puede ser compartida, almacenada y analizada de manera consistente [22].

Cyber Observable eXpression (CybOX) es un lenguaje XML usado para la representación y el intercambio de *cyber observables*. Se puede definir un *cyber observable* como un evento o propiedad con estado en un contexto informático [23].
CybOX provee la forma para caracterizar y evaluar amenazas detallando patrones de ataque, caracterizar malware, forensia digital, respuesta a incidentes, entre otros.

STIX aprovecha CybOX para describir eventos o propiedades con estado en un contexto informático.
Para desarrollar herramientas que se beneficien de este lenguaje el proyecto STIX [25] provee librerías Python y Java.

DFXML

Digital Forensics XML (DFXML) es un lenguaje XML usado para el procesamiento automatizado del análisis digital. DFXML contiene información tanto de los resultados del procesamiento forense como de las herramientas utilizadas para realizar dicho procesamiento. Actualmente no existe un estándar DFXML y no hay ningún esquema fijo, pero existe un borrador de esquema hecho por el National Software Reference Library (NSRL) [26, 27].

Algunas herramientas que producen DFXML

- AFXML convierte la metadata de las imágenes de los discos en formato DFXML.
- EWFINFO, parte de libewf, puede exportar metadata de imágenes de discos EWF en formato DFXML.

Algunas herramientas que consumen DFXML

- IMAP.PY, parte de la distribución fiwalk, dibujará un mapa de lo que hay en un disco. Solo es útil para particiones chicas.
- IVERIFY.PY, parte de la distribución fiwalk, verificará que el contenido de los archivos en el archivo DFXML no ha sido modificado.

DFXML Toolkit Las siguientes toolkits son útiles para la creación de nuevas herramientas que lean y escriban DFXML:

- DFXML.PY es un módulo Python implementa objetos para la lectura y escritura de DFXML.
- XML.CPP y XML.H están incluidos en bulk_extractor y md5deep (versión 4) y sirve para la generación de DFXML.
- XML.C y XML.H están incluidos en la nueva versión de photorec y sirve para la generación de DFXML.

XIRAF [28]
Es utilizado para gestionar y consultar las trazas extraídas de la evidencia digital. La arquitectura XIRAF indexa las imágenes de discos y las almacena en formato XML, esto permite utilizar un sencillo pero potente lenguaje para realizar consultas, el XQuery. Además provee un entorno de consultas enriquecido donde el usuario puede navegar, buscar y utilizar plantillas predeterminadas. Sin embargo, esta arquitectura está diseñada sólo para indexar y recuperar la evidencia digital y no admite ningún medio para combinar información de varios tipos. Además, la arquitectura carece de flexibilidad para extraer el contenido que pueden proporcionar correlaciones fundamentales en datos de otras partes. [29]

2.5.2. Modelos

Modelo para sistemas de detección de intrusiones

A continuación se presenta el modelo para sistemas de detección de intrusiones (IDS) en tiempo real propuesto por Dorothy E. Denning en "An Intrusion-Detection Model" [1] que tiene como objetivo detectar un amplio rango de violaciones de seguridad. Si bien este modelo está concebido para IDS, es independiente de cualquier sistema particular, entorno de aplicación, vulnerabilidad de sistema o tipo de intrusión, proporcionando, de este modo, un framework para un sistema de detección de intrusiones de propósito general. Estas características lo posicionan, a nuestro criterio, como un insumo interesante de considerar en el contexto de nuestro trabajo.

Componentes del modelo

Sujetos: Iniciadores de actividades en un sistema. *Ej.: usuarios.*
Objetos: Recursos administrados por el sistema. *Ej.: archivos, comandos, dispositivos.*
Registros de auditoría: Registro que se genera cuando se importan datos de una fuente que almacena las acciones que los sujetos han realizado sobre los objetos. *Ej.: importación del log de recepción de llamadas, importación del log de envío de mensajería instantánea.*
Se representa con la 6-tupla:

```
<Sujeto, Acción, Objeto, Condición de Excepción, Uso del Recurso, Marca de tiempo>
```

Los elementos de la misma que no fueron explicados anteriormente se detallan a continuación:

- **Acción:** Operación realizada por un Sujeto sobre o con un Objeto.
- **Condición de Excepción:** Indica, si existe alguna, condición de excepción. Esta debería ser la condición de excepción actual levantada por el sistema y no una aparente condición de excepción devuelta por un sujeto.
- **Uso del Recurso:** Duración de tiempo que se utilizó el recurso en caso de que se pueda medir.
- **Marca de tiempo:** Marca de fecha y hora que identifica cuando la acción fue llevada a cabo.

Perfiles: Estructuras que caracterizan el comportamiento de sujetos con respecto a objetos en términos de métricas y modelos estadísticos de la actividad observada. Los perfiles son generados automáticamente a partir de templates, constituyendo su creación, el primer paso en el proceso de análisis de la evidencia.

La estructura de los perfiles viene dada por la 10-tupla definida a continuación:

```
<Nombre de variable, Patrón de acción, Patrón de excepción, Patrón de uso del recurso, Período, Tipo de
variable, Umbral, Patrón de Sujeto, Patrón de Objeto, Valor>
```

Cada elemento se describe debajo:

- **Nombre de variable:** Es el nombre de la variable utilizada.
- **Patrón de acción:** Patrón que matchea con cero o más acciones en los registros de auditoría.
- **Patrón de excepción:** Patrón que matchea con el campo "Condición de Excepción" de un Registro de Auditoría.
- **Patrón de uso del recurso:** Patrón que matchea con el campo "Uso del Recurso" de un Registro de Auditoría.
- **Período:** Intervalo de tiempo entre medidas. Nulo significa que el intervalo de tiempo no es fijo, (el período es la duración de la actividad).
- **Tipo de variable:** Nombre que define la métrica y el modelo estadístico usado.
- **Umbral**: Parámetros que definen los límites utilizados en los tests estadísticos para determinar anomalías.
- **Patrón de Sujeto:** Patrón que matchea con el campo "Sujeto" de un Registro de Auditoría.
- **Patrón de Objeto:** Patrón que matchea con el campo "Objeto" de un Registro de Auditoría.
- **Valor:** Valor más reciente de observación y los parámetros utilizados por el modelo estadístico para representar la distribución de los valores anteriores.

Los templates tienen un formato similar a los perfiles a excepción del sujeto y el objeto que son definidos a partir de patrones.

```
<Nombre de variable, Patrón de acción, Patrón de excepción, Patrón de uso del recurso, Período, Tipo de variable, Umbral,
        matching-pattern ← replacement-pattern, matching-pattern ← replacement-pattern, Valor inicial>
```

Métricas: Una métrica es una variable aleatoria x que representa una medida cuantitativa acumulada a lo largo de un período de tiempo. Se definen tres tipos de métricas:

1. **Contador de eventos:** x es el número de registros de auditoría que satisfacen alguna propiedad en algún período, donde cada registro de auditoría se corresponde con un evento. *Ejemplos: Cantidad de logins en una hora, cantidad de intentos fallidos durante un minuto.*
2. **Tiempo entre eventos:** x es el tiempo entre dos eventos relacionados. *Ejemplo: El tiempo entre logins sucesivos para una misma cuenta de usuario.*
3. **Medición de recursos:** x es la cantidad de recurso consumido por alguna acción durante un período. *Ejemplo: Total de páginas impresas por usuario por día.*

Modelos estadísticos: Dados, una métrica para una variable aleatoria x y n observaciones (x_1, ..., x_n) se considera un modelo estadístico de x cuyo propósito es determinar si una nueva observación x_{n+1} es anormal con respecto a las observaciones anteriores.

Se proponen los siguientes modelos estadísticos:

1. **Modelo operacional:** Se fijan límites y dada una observación x_{n+1} se clasifica la misma según si cae dentro del intervalo definido o si cae por fuera del mismo.
2. **Media y desviación estándar:** Se calcula la media y la desviación estándar de las observaciones x_1, ..., x_n y dada la observación x_{n+1} se clasifica la misma según si cae dentro del intervalo de confianza o fuera. El intervalo de confianza se define como $media \pm d \times stdev$ siendo d un parámetro dado. La media y la desviación estándar se calculan como sigue:

$$sum = x_1 + \ldots + x_n$$

$$sumsquares = x_1^2 + \ldots + x_n^2$$

$$mean = \frac{sum}{n}$$

$$stdev = \sqrt{(\frac{sumsquares}{n-1} - mean^2)}$$

   Este modelo se aplica con métricas del tipo contadores de eventos, tiempo entre eventos y medición de recursos, acumulado a lo largo de un intervalo de tiempo fijo o entre dos eventos relacionados.

La aplicación de este modelo puede ser observada en el estudio del registro de auditoría generado por los SMS, por ejemplo, si el sujeto habitualmente envía una cantidad x de mensajes de texto a un contacto por día, detectar que en un día se han enviado una cantidad que cae por fuera del intervalo de confianza para esta métrica sería sospechoso.

3. **Modelo multivariado:** Es similar al modelo de media y desviación estándar, a diferencia de que se basa en la correlación entre dos o más métricas. Es usado cuando existe mayor poder de información a raíz de la correlación entre las métricas, que al estudiar las mismas por separado.
   La aplicación de este modelo se puede observar en el estudio del registro de auditoría de llamadas debido a la posibilidad de correlacionar la información medida con dos métricas distintas. En el contexto de este trabajo se correlacionan la cantidad de llamadas con la duración de las mismas, por ejemplo, observar una llamada sola quizás, a priori, puede no decir nada, pero si la misma tiene una duración que cae fuera del intervalo determinado por la media y la desviación estándar, entonces podría ser relevante.
4. **Modelo de proceso de Markov:** Aplica solo con una métrica del tipo contador de eventos. Utiliza una matriz de transiciones de estados para caracterizar la frecuencia de transiciones entre estados. Una observación x_{n+1} se clasifica según si la probabilidad es baja (o alta) de cambiar del estado anterior al estado actual.
5. **Modelo de series de tiempo:** Utiliza un intervalo de tiempo junto con una métrica del tipo contador de evento o medición de recurso. Toma en cuenta el orden y los tiempos entre apariciones de las observaciones $x_1, \ldots, x_n$ como también sus valores. Una observación x_{n+1} se clasifica como anormal si la probabilidad de su ocurrencia en el tiempo dado es muy baja.

Registros de anomalías: Son generados cuando un comportamiento anormal es detectado. La detección de comportamiento anormal es una funcionalidad dentro del proceso de "clasificación de la evidencia".

Reglas de actividad: Son acciones que se deben tomar cuando una condición es satisfecha, actualizan perfiles, detectan comportamiento anormal, relacionan anomalías con posibles incidentes y generan reportes. Estas reglas se lanzan cuando se ejecutan algún tipo de análisis o procesamiento de la evidencia.

3. Análisis

3.1. Objetivo

Según lo expuesto en el estado del arte, donde se comentan diferentes formas de análisis y lenguajes estándares para la representación de la información, el objetivo principal es desarrollar una herramienta que sirva como asistente a un investigador forense, para detectar anomalías en la operativa de un dispositivo móvil. Se entiende como asistente un proceso que, dada cierta evidencia, indica donde hacer foco para avanzar en la investigación.

Dicha herramienta permitirá incorporar diversos tipos de análisis y será capaz de obtener insumos para el análisis desde información expresada en lenguajes estándares.
En este contexto, el modelo propuesto por Dorothy E. Denning es de gran interés, el mismo será adaptado y utilizado como parte del core de la herramienta a desarrollar.

En las secciones subsiguientes se describen los requerimientos de la herramienta, se analizan los diferentes componentes de Android y su correspondencia con el modelo de referencia, además se hace una validación del modelo en este contexto, se considera el manejo del huso horario y la auditoría del sistema.

3.2. Herramienta a desarrollar

3.2.1. Requerimientos Funcionales

1. Detección de comportamiento anómalo (sospechoso).
   a) Estudiar el registro de SMSs.
   b) Estudiar el registro de llamadas.
   c) Estudiar el registro de emails.
   d) Analizar conjuntamente emails, llamadas y SMSs.
   e) Correlacionar eventos con distintas métricas, por ejemplo, las llamadas utilizando las métricas contador de eventos y uso de recurso.
2. Graficar histograma donde se muestren la cantidad de comunicaciones entre el propietario y los contactos.
   a) Generar una gráfica del tipo histograma donde se muestre la cantidad de comunicaciones que fueron realizadas.
   b) Obtener un desglose de las comunicaciones por tipo, por ejemplo SMS, email y llamadas.
   c) Permitir acotar los datos considerados a través de la definición de un rango de fechas.
3. Brindar un mecanismo para la normalización de los sujetos.
4. Exportación de resultados en formato CSV.
5. Guardar el estado del análisis.

3.2.2. Requerimientos No Funcionales

1. Utilizar como referencia el modelo propuesto por Dorothy E. Denning.
2. Arquitectura extensible que permita el agregado de componentes para la carga de datos desde distintas fuentes, distintos tipos de análisis y la exportación de los resultados para que éstos sean usados por otras herramientas.

3.2.3. Casos de uso

Actores del sistema

Investigador: Es quién se encarga de realizar la carga de datos, realizar el análisis de los mismos y de exportarlos en caso de ser necesario. Es el actor principal y encargado de llevar a cabo los casos de uso *Búsqueda de comportamiento anómalo*, *Ver contactos*, *Carga de datos*, *Ver comunicaciones* y *Exportar datos*.

Proceso automático: Los sujetos, los objetos y los plugins se cargan automáticamente en el sistema. Los plugins son cargados cuando se ejecuta la aplicación y cuando se modifica el directorio que los contiene, en cambio los sujetos y los objetos, son dados de alta mientras se lleva a cabo el caso de uso *Carga de datos*.

Diagrama de casos de uso y presentación de actores

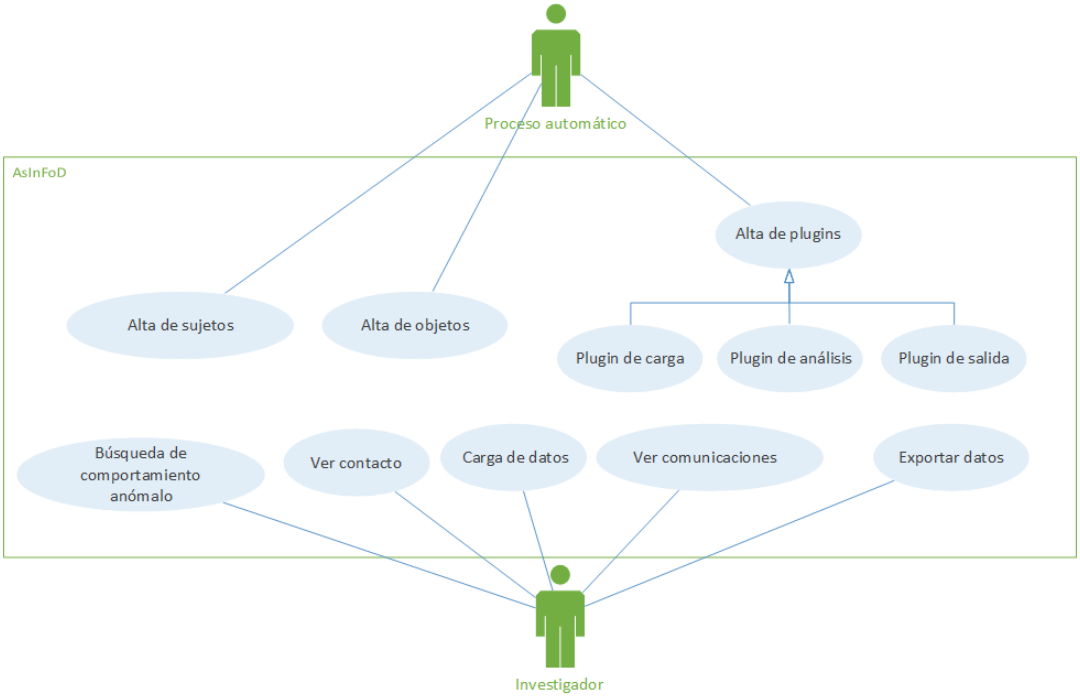

Figura 8: Casos de uso y Actores

Casos de uso de carga de datos

1. **Carga de datos**

Nombre:	Carga de datos
Actores:	Investigador
Descripción:	El caso de uso comienza cuando el investigador ingresa la fuente de los datos que desea cargar. El investigador invoca la operación que desea ejecutar para realizar la carga de los datos y luego de la misma el sistema queda con los datos cargados.
Precondiciones:	El sistema ya cuenta con algún plugin de carga.
Postcondiciones:	El sistema queda con datos en sus estructuras.
Flujo principal:	**1.** El investigador ingresa un identificador de la fuente (número telefónico ó referencia). **2.** El investigador elige el archivo con los datos que desea cargar. **3.** El investigador elige la operación que desea ejecutar. Los pasos 2 y 3 se repiten mientras hayan archivos por ser cargados. **4.** El investigador selecciona Cargar datos.

Tabla 2: CU: Carga de datos

Casos de uso de análisis

1. **Búsqueda de comportamiento anómalo**

Nombre:	Búsqueda de comportamiento anómalo
Actores:	Investigador
Descripción:	El caso de uso comienza cuando el investigador elige la opción analizar. El sistema ofrece la posibilidad de acotar el universo de datos sobre los que se aplicará el análisis.
Precondiciones:	-
Postcondiciones:	-
Flujo principal:	**1.** El investigador acota el universo de datos sobre los que se aplicará el análisis a través de la aplicación de un filtro. **2.** El investigador elige la opción “Analizar”. **3.** El investigador elige la operación que desea ejecutar. **4.** Al terminar el sistema indica (en caso de que exista) cuáles son los incidentes detectados como sospechosos para que el investigador refuerce la atención en dichos eventos.

Tabla 3: CU: Búsqueda de comportamiento anómalo

2. **Ver comunicaciones**

Nombre:	Ver comunicaciones
Actores:	Investigador
Descripción:	El caso de uso comienza cuando el investigador elige la opción ver comunicaciones. El sistema desplegará un histograma contabilizando las comunicaciones llevadas a cabo con los contactos elegidos.
Precondiciones:	Hay al menos algún contacto seleccionado
Postcondiciones:	Se genera una histograma
Flujo principal:	**1.** El investigador elige la opción “Ver comunicaciones”, el sistema despliega un histograma contabilizando las comunicaciones llevadas a cabo con los contactos elegidos.
Flujos Alternativos:	**1a.** El investigador elige un rango de fechas.

Tabla 4: CU: Ver historial de comunicaciones

3. **Ver contactos**

Nombre:	Ver contactos
Actores:	Investigador
Descripción:	El caso de uso comienza cuando el investigador elige la opción ver contactos. El sistema desplegará toda la información de los contactos.
Precondiciones:	-
Postcondiciones:	Se despliega la información de contactos
Flujo principal:	**1.** El investigador elige la opción "Ver contactos", el sistema mostrará nombre completo, números de teléfonos, dirección de correo (si cuenta con alguna) y la ruta donde se encuentra la foto (en caso de que haya) de los contactos.

Tabla 5: CU: Ver contactos

4. **Ver análisis**

Nombre:	Ver análisis
Actores:	Investigador
Descripción:	El caso de uso comienza cuando el investigador elige la opción ver análisis. El sistema desplegará la lista de los análisis realizados y el ivestigador puede seleccionarlos para ver sus detalles.
Precondiciones:	Hay algún análisis seleccionado.
Postcondiciones:	Se despliega la información del análisis seleccionado.
Flujo principal:	**1.** El investigador elige la opción "Ver análisis", el sistema mostrará una lista con los análisis llevados a cabo. **2.** El investigador elige alguno de los análisis listado y puede ver los detalles del mismo.

Tabla 6: CU: Ver análisis

Casos de uso de configuración

1. **Alta de Sujetos**

Nombre:	Alta de Sujetos
Actores:	Proceso automático
Descripción:	El caso de uso comienza cuando un plugin de carga realiza la carga de registros en el sistema.
Precondiciones:	-
Postcondiciones:	Se generan los nuevos Sujetos.
Flujo principal:	**1.** Si el sujeto no existe en el sistema se da alta automáticamente.

Tabla 7: CU: Alta de Sujetos

2. **Alta de Objetos**

Nombre:	Alta de Objetos
Actores:	Proceso automático
Descripción:	El caso de uso comienza cuando un plugin de carga realiza la carga de registros en el sistema.
Precondiciones:	-
Postcondiciones:	Se generan los nuevos Objetos para ser utilizados.
Flujo principal:	**1.** Si el objeto no existe en el sistema se da alta automáticamente.

Tabla 8: CU: Alta de Objetos

3. **Alta de plugins**

Nombre:	Alta de plugins
Actores:	Proceso automático
Descripción:	El sistema carga los plugins que se encuentran en el directorio destinado a los plugins.
Precondiciones:	-
Postcondiciones:	El plugin queda cargado en el sistema
Flujo principal:	**1.** El sistema chequea por cambios en el directorio destinado a los plugins. Si hay cambios el plugin es cargado al sistema y se lo deja disponible para su utilización.

Tabla 9: CU: Alta de plugins

Casos de uso de salida

1. **Exportar datos**

Nombre:	Exportar datos
Actores:	Investigador
Descripción:	Se exportan los datos filtrados.
Precondiciones:	-
Postcondiciones:	Se exporta a un archivo con los datos filtrados.
Flujo principal:	**1.** El investigador elige el filtro a aplicar. **2.** El investigador elige la opción "Exportar datos" y selecciona el nombre del archivo y la operación que desea utilizar.

Tabla 10: CU: Exportar datos

3.3. Insumos de Android para el análisis

3.3.1. Ubicación de los insumos en Android

Los insumos que Android provee se almacenan en bases de datos SQLite. Dado que no existe un estándar que defina donde guardar la información, los fabricantes de smartphones tienen la libertad de elegir en donde almacenar los datos. Un caso particular es el de Samsung que almacena las llamadas en una tabla diferente a la mayoría de los fabricantes [30] [31]

```
/data/data/com.sec.android.provider.logsprovider/databases/logs.db
```

En el caso de las aplicaciones de terceros, la ubicación de sus bases de datos va a depender del fabricante de las mismas. Por lo dicho anteriormente, no se pueden definir ubicaciones desconociendo el entorno de donde se las extraerán.
A continuación, se muestra una tabla con algunas de las posibles ubicaciones donde encontrar algunos insumos de interés.

Registros de auditoría	**Representación en Android**	
	Tipo	**Ubicación**
Agenda telefónica	SQLite	`/data/data/com.android.providers.contacts/databases/contacts2.db`
Historial de llamadas	SQLite	`/data/data/com.android.providers.contacts/databases/contacts2.db` `/data/data/com.sec.android.provider.logsprovider/databases/logs.db`
SMS / MMS	SQLite	`/data/data/com.android.providers.telephony/databases/mmssms.db`
Historial de ubicación	SQLite	`/data/com.google.android.apps.maps/databases/da_destination_history` `/data/com.google.android.apps.maps/databases/search_history.db`

Tabla 11: Ubicación en Android de los insumos

3.3.2. Bases de datos mmssms.db, logs.db y contacts2.db

Se notó la falta de documentación oficial por parte de Android acerca de las bases de datos del sistema, básicamente lo que se podía leer en páginas oficiales hacía referencia a las APIs con las cuales programar pero no se explicaban las estructuras de las bases de datos, su semántica ni como se relacionaban entre ellas. Por tal motivo se optó por conseguir las bases de datos donde se almacenan los mensajes de texto, las llamadas y los contactos [5] para investigarlas a modo de interpretar tanto el contenido como sus relaciones.

SMSs, hilos y contactos

Los SMSs se los encuentra en la base de datos mmssms.db, específicamente en la tabla `sms`. En esta tabla se destaca la particularidad del campo `address`, este campo es donde se almacena *"con quién se intercambió el mensaje"* y puede ser un texto o un número de teléfono con o sin código de país, esto implica que necesariamente se deberá proveer un mecanismo de normalización de los datos.

Consultando todos los mensajes intercambiados con un número (normalizado) se observa que el campo `thread_id` puede variar, esto se debe a como Android organiza el flujo de mensajes. Para cada número contactado utiliza un único `thread` tanto para agrupar los mensajes enviados como los recibidos y para cada grupo de destinatarios utiliza un único `thread` por el cual agrupa cada mensaje enviado al grupo; los mensajes generados como respuesta a mensajes enviados a un grupo se agregan al `thread` correspondiente al número que envía la respuesta, **no** al `thread` del grupo.

Gráficamente se lo puede ver de la siguiente manera:

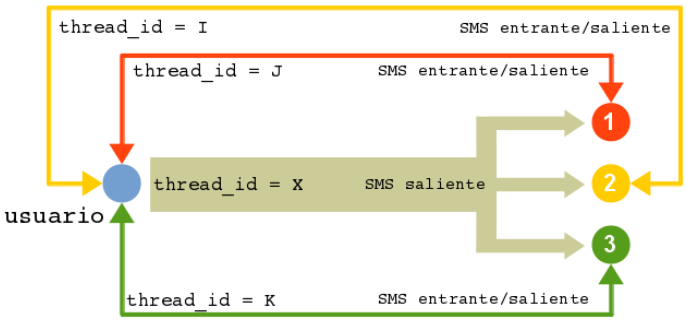

Figura 9: Asociación de threads

Este mecanismo de *threads* puede proporcionar vínculos entre números de teléfono simplemente observando las pertenencias a grupos a través de la coincidencia del valor del campo `thread_id`, por ejemplo:

"Un investigador puede filtrar por un número de su interés y consultar por el valor del campo ***`thread_id`****, luego para cada valor distinto del campo* ***`thread_id`*** *puede obtener todos los mensajes enviados a esos* ***`threads`*** *y quedarse con los distintos valores del campo* ***`address`****"*

De esta forma obtiene los números de teléfonos vinculados entre sí.

Para obtener el nombre con los cuáles se agendaron los números hay que buscar cuál es el valor del campo `person` en la tabla `sms`, ese campo es una clave foránea de la tabla `raw_contacts` que pertenece a la base de datos `contacts2.db`. Una particularidad del campo `person` es que para un mismo número de teléfono éste puede valer `null` o puede tener la clave foránea de la tabla `raw_contacts`, por lo cual se deben descartar aquellos registros cuyo valor del campo `person` sea `null`.

[5] El smartphone de donde se obtuvieron las bases de datos es un Samsung GT-I8160L

Luego de haber obtenido el valor de la clave foránea de la tabla `raw_contacts`, es cuestión de consultar por los valores de los campos `display_name` y `display_name_alt` correspondientes al registro cuya clave se halló ya que los mismos almacenan los nombres con los cuales se agendó el contacto.

Analizando la tabla `logs` de la base de datos `logs.db` se observa que la misma almacena varios tipos de logs, para obtener todas las llamadas realizadas, recibidas y perdidas se debe filtrar según el logtype=100[32]

Adicionalmente, al momento de realizar un análisis exhaustivo de un contacto es de ayuda poder contar con la foto del mismo, en caso de que exista la misma se la puede encontrar en:

```
'content://com.android.contacts/contacts/'||contacts._id||'/photo'
  'content://com.android.contacts/display_photo/'||photo_file_id
```

Donde `contacts._id` y `photo_file_id` son respectivamente el identificador del contacto y el identificador de la foto, ambos campos se encuentran en la tabla `contacts` de la base de datos `contacts2.db`.

3.4. Mapeo entre el modelo y Android

En la tabla 12, con el fin de generar una mayor comprensión, se brinda un mapeo preliminar en alto nivel de como se puede realizar la correspondencia entre el modelo propuesto por Dorothy E. Denning y el dominio dado en un smartphone con Android.

Modelo	**Android**
Sujetos	Sistema operativo, procesos, malwares, etc.
Objetos	Agenda telefónica, Aplicaciones (de redes sociales, de mensajería instantánea, de navegación web, de reproducción de contenido multimedia, etc), archivos, etc.
Acciones	Envío y recepción de SMSs y llamadas, etc.
Registros de auditoría	Historial de llamadas, Registros de SMS / MMS, Historial de navegación, etc.

Tabla 12: Mapeo entre el modelo y Android

Mapeo entre los registros de auditoría del modelo y los logs de Android
Para fijar ideas, en la tabla 13 se muestra el mapeo entre el registro de auditoría del modelo y los logs del *Historial de llamadas* y *SMS* correspondientes al dispositivo utilizado en la sub-sección 3.3.2.

En la columna *Registro de auditoría* se especifica el tipo de log al que se hace referencia.
Las columnas *Sujeto*, *Acción*, *Uso de recurso* y *Marca de tiempo* contienen los nombres de los campos de las bases de datos correspondientes. La columna *Objeto* contiene la interpretación que se hace del contenido del log.

Registros de auditoría	**Sujeto**	**Acción**	**Objeto**	**Uso del recurso**	**Marca de tiempo**
Historial de llamadas [6]	number	type [7]	llamadas	duration [8]	date [9]
SMS [10]	address	type [11]	SMS	N/A	date

Tabla 13: Mapeo entre los registros de auditoría del modelo y los logs de Android

[6] Base contacts2.bd tabla calls y en Samsung base logs.db tabla logs
[7] 1=llamada recibida, 2=llamada realizada
[8] En segundos
[9] Unixtimestamp en milisegundos
[10] Base mmssms.bd tabla sms
[11] 1=SMS entrante, 2=SMS saliente

Ejemplos de Templates [12]

Llamadas entrantes

<contador_llamada_entrante, 1, NULL, 1, DAY, CONTADOR_DE_EVENTOS, 5, * → number ← number, LLAMADAS, 0>

<durac_llamada_entrante, 1, NULL, duration, NULL, MEDICION_DE_RECURSOS, 5, * → number ← number, LLAMADAS, 0>

Llamadas saliente

<contador_llamada_saliente, 2, NULL, 1, DAY, CONTADOR_DE_EVENTOS, 5, * → number ← number, LLAMADAS, 0>

<durac_llamada_saliente, 2, NULL, duration, NULL, MEDICION_DE_RECURSOS, 5, * → number ← number, LLAMADAS, 0>

SMS entrantes

<contador_SMS_entrantes, 1, NULL, N/A, DAY, CONTADOR_DE_EVENTOS, 20, * → address ← address, SMS, 0>

SMS salientes

<contador_SMS_salientes, 2, NULL, N/A, DAY, CONTADOR_DE_EVENTOS, 20, * → address ← address, SMS, 0>

3.5. Validación del modelo en el contexto del proyecto

El modelo de Dorothy E. Denning fue concebido como un modelo de un sistema experto de detección de intrusiones en tiempo real capaz de detectar robos, penetraciones y otras formas de abuso informático. Los modelos estadísticos en él propuestos se mantienen vigentes, esto puede verse en el paper "A Review of Anomaly based IntrusionDetection Systems" [33] fechado en setiembre de 2011.

La gran diferencia entre la realidad supuesta por Dorothy E. Denning y la de este proyecto está en que en el primer caso la detección es en tiempo real y en el segundo se cuenta con el universo de datos de estudio. Lo dicho anteriormente sumado a los tipos de modelos estadísticos propuestos supone realizar, al menos, una mínima validación de la aplicabilidad del modelo propuesto en el nuevo contexto.

De esos modelos estadísticos[13], a priori, parecen más apropiados en la detección de comportamiento sospechoso para las comunicaciones los modelos *Media y desviación estándar, Multivariado y Series de tiempo*. Para esta validación se usarán los dos primeros y en caso de que nos sean útiles se considerarán los restantes comenzando por *Series de tiempo*.

En tal sentido se definirán los siguientes casos de prueba, donde el objetivo es detectar comportamientos sospechosos:

- Análisis de cantidad de SMSs enviados/recibidos por día asociados a un contacto.
- Análisis de cantidad de LLAMADAS enviadas/recibidas por día asociadas a un contacto.
- Análisis de cantidad de SMSs y LLAMADAS enviados/recibidos por día asociados a un contacto.

El conjunto de datos a usar en los casos de prueba fue generado por el grupo. Para cada caso se presenta una tabla con los valores correspondientes a la cantidad de comunicaciones intercambiadas entre el usuario del teléfono y un contacto en un período de 29 días consecutivos.

[12] Estructura: <Nombre de variable, Patrón de acción, Patrón de excepción, Patrón de uso del recurso, Período, Tipo de variable, Umbral, matching-pattern ← replacement-pattern, matching-pattern ← replacement-pattern, Valor inicial>

[13] *Modelos: Operacional, Media y desviación estándar, Multivariado, Procesos de Markov y Series de tiempo*

Análisis de la cantidad de SMSs enviados/recibidos por día asociados a un contacto

En la tabla 14 se presentan los datos a analizar, los cuales corresponden a SMSs y están totalizados por día.

Día	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
# SMSs	3	6	5	2	4	5	2	1	2	3	12	0	2	4	2
Día	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	–
# SMSs	7	0	0	0	16	5	7	6	4	18	10	10	20	4	–

Tabla 14: Cantidad de SMSs enviados/recibidos por día asociados a un contacto

Al analizar los datos usando la métrica *contador de eventos* y el modelo estadístico *media y desviación estándar* se obtienen como comunicaciones sospechadas de comportamiento anómalo las que corresponden a los días 1, 2, 11 y 20, los que aparecen destacados en rojo en la figura 10.
Además se muestra la cantidad de SMSs intercambiados por día (amarillo) y el intervalo de confianza (fuera de la franja delimitada por las líneas azules se fijan los sospechosos) :

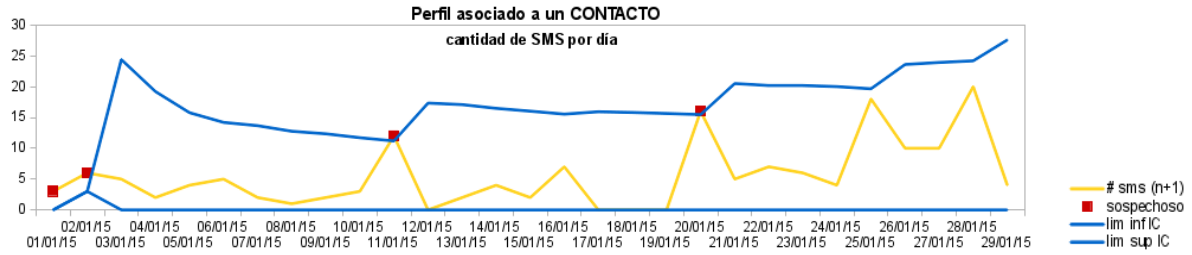

Figura 10: Detección de anomalías analizando SMSs

Análisis de la cantidad de LLAMADAS enviadas/recibidas por día asociadas a un contacto

En la tabla 15 se presentan los datos a analizar, los cuales corresponden a llamadas y están totalizadas por día.

Día	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
# LLAMADAS	1	2	3	4	3	2	3	5	3	2	4	2	4	6	4
Día	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	–
# LLAMADAS	3	2	4	6	7	2	1	2	6	10	6	11	4	6	–

Tabla 15: Cantidad de LLAMADAS enviadas/recibidas por día asociadas a un contacto

Al analizar los datos usando la métrica *contador de eventos* y el modelo estadístico *media y desviación estándar* se obtienen como comunicaciones sospechadas de comportamiento anómalo las que corresponden a los días 1 y 2, los que aparecen destacados en rojo en la figura 11.
Además se muestra la cantidad de LLAMADAS intercambiadas por día (amarillo) y el intervalo de confianza (fuera de la franja delimitada por las líneas azules se fijan los sospechosos) :

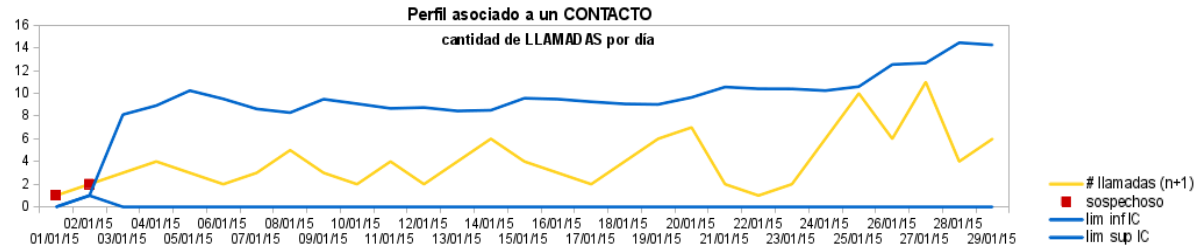

Figura 11: Detección de anomalías analizando LLAMADAS

Análisis de la cantidad de SMSs y LLAMADAS enviados/recibidos por día asociados a un contacto

En la tabla 16 se presentan los datos a analizar, los cuales corresponden a SMSs y llamadas y están totalizados por día.

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# SMSs	3	6	5	2	4	5	2	1	2	3	12	0	2	4	2
# LLAMADAS	1	2	3	4	3	2	3	5	3	2	4	2	4	6	4

Día	16	17	18	19	20	21	22	23	24	25	26	27	28	29	–
# SMSs	7	0	0	0	16	5	7	6	4	18	10	10	20	4	–
# LLAMADAS	3	2	4	6	7	2	1	2	6	10	6	11	4	6	–

Tabla 16: Cantidad de SMSs y LLAMADAS enviados/recibidos por día asociados a un contacto

Al analizar los datos usando la métrica *contador de eventos* y el modelo estadístico *media y desviación estándar* se obtienen como comunicaciones sospechadas de comportamiento anómalo las que corresponden a los días 1, 2, 11, 20 y 25, los que aparecen destacados en rojo en la figura 12.
Además se muestra la cantidad de SMSs y LLAMADAS intercambiados por día (amarillo) y el intervalo de confianza (fuera de la franja delimitada por las líneas azules se fijan los sospechosos) :

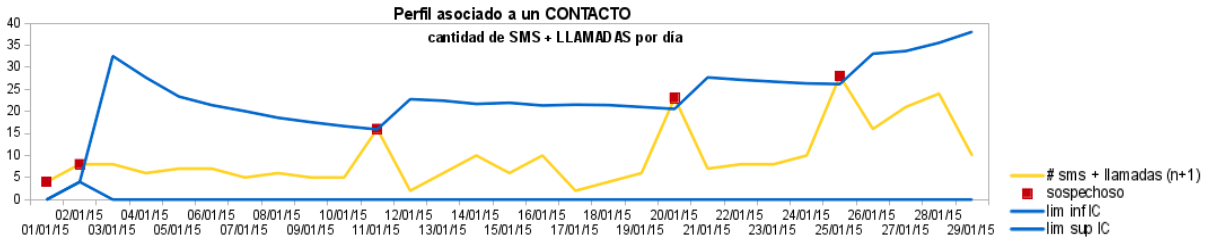

Figura 12: Detección de anomalías analizando SMSs y LLAMADAS

Conclusiones

Los ejemplos presentados representan un amplio espectro de casos donde aplicar un análisis que permita detectar comportamiento sospechoso ya que:

- Los objetos de estudio, LLAMADAS y SMSs, caracterizan objetos que al aplicarles una acción generan, respectivamente, eventos con y sin duración.
- El criterio de agrupamiento usado para los eventos registrados (insumos para el análisis) es el tiempo, en este caso por día, pero el análisis también puede realizarse por hora, semana, mes, año, etcétera.

Utilizando el modelo estadístico *media y desviación estándar* se detectaron días con actividad fuera del comportamiento típico, esto puede comprobarse rápidamente observando la curva amarilla de las gráficas.

El uso del modelo estadístico *multivariado* permitió detectar días con actividad fuera del comportamiento típico no detectados al usar media y desviación estándar. La diferencia entre estos análisis está en que para *media y desviación estándar* se analizaron los SMSs por un lado y las LLAMADAS por otro, y en el *multivariado* se correlacionaron dichos datos; notar que en el último análisis se usaron los datos de los anteriores.

Si bien hay diferencias entre el dominio de aplicación para el que fue pensado el modelo de Dorothy E. Denning y el que se estudia en este proyecto, dado el nivel de abstracción con que fueron definidos los conceptos que maneja y la factibilidad de uso vista con los ejemplos anteriormente planteados, se considera apropiado su utilización en el marco de este trabajo.

3.6. Importancia del huso horario

Cuando se registra un evento uno de sus datos relevantes es la marca de tiempo (la que típicamente se registra como un número de serie a partir de una fecha), asociado a ésta y que no se suele registrar (al menos no hemos encontrado casos donde se registre) se encuentra el huso horario.

Al momento de analizar los eventos ocurridos es de suma importancia tener en cuenta el huso horario, ya que de no hacerlo se puede llegar a conclusiones incorrectas, para ver de una manera más clara su impacto se presenta el siguiente ejemplo.
Supongamos que se están analizando todos los SMS que están en la bandeja de enviados de todos los empleados de una compañía correspondientes al día 28 de Agosto de 2014, día en que se sabe fue filtrada información confidencial a la competencia. La marca de tiempo se registra como un número en milisegundos a partir del 1/1/1970 0:00:00 horas GMT (Greenwich Mean Time), por lo que si el SMS que contiene la información filtrada se envió el 28/08/2014 a las 21:40:29 horas de Uruguay el valor correspondiente será 1409272829000 [14] respecto de GMT.
Al determinar el intervalo de milisegundos que hay entre las 0:00:00 y 23:59:59 horas del día 28/08/2014 se buscaría en aquellos registros que tienen marca de tiempo entre los valores 1409184000000 y 1409270399000. Como se aprecia este intervalo excluye el registro con el elemento probatorio. El problema está en que el 28/08/2014 a las 21:40:29 horas de Uruguay en GMT corresponde al 29/08/2014 a las 0:40:29 horas y para la determinación del intervalo no se tuvo en cuenta el huso horario.

[14] se puede verificar este dato, dividiéndolo entre 1000 para que quede en segundos e ingresando ese resultado a http://coderstoolbox.net/unixtimestamp/

3.7. Auditoría del sistema

Teniendo como motivador el contexto forense, para garantizar que el análisis realizado de los datos, desde que estos ingresan a la herramienta hasta que se obtienen los resultados, puede reproducirse obteniendo para un mismo conjunto de datos iniciales a través de la misma secuencia de pasos los mismos resultados, se proveerá un módulo que auditará los eventos que provoquen cambios en el sistema. Es decir que tanto al momento de realizar la carga de los datos como el análisis de los mismos se debe generar un registro que sea lo suficientemente descriptivo con el fin de poder reproducir el proceso sin lugar al error.
Datos importantes para ser registrados son: la operación, el momento que se realizó la misma y los parámetros que se utilizaron.

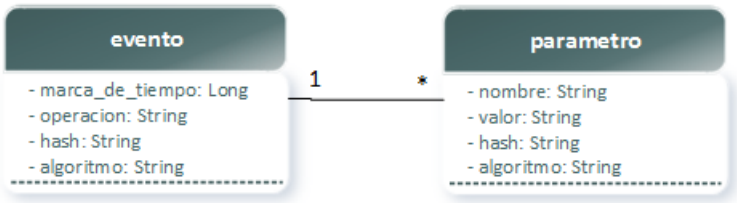

Figura 13: Modelo de dominio para el módulo de auditoría

3.8. Modelo de dominio

Se presentarán dos modelos de dominio, correspondiendo uno a lo que puede considerarse el core de la aplicación y otro a la extensión que se encargará de detectar el comportamiento sospechoso, contemplando de esta manera los requerimientos de extensibilidad y detección de comportamiento sospechoso.
Para éstos se dará una breve explicación de las entidades y vínculos más relevantes para el desarrollo del proyecto.

Como punto a destacar se tiene que el resultado de un análisis se modelará como un registro de auditoría, la consecuencia de hacer esto es que un resultado puede ser analizado, en otras palabras, se dota a la herramienta con la capacidad de hacer re-análisis de los eventos registrados.

3.8.1. Core

Entidades asociadas al modelo de referencia [15]
registro_de_auditoria: Representa al registro de auditoría y es donde estarán los registros de los eventos que se analizarán.
sujeto: Representa al sujeto, que es quién hace una acción sobre un objeto, en este contexto puede ser, por ejemplo un *número de teléfono* o una *dirección de correo electrónico*.
objeto: Representa al objeto, que es sobre el que se realiza la acción, un ejemplo es el objeto *SMS*.
accion: Representa a la acción, que es la que se hace por parte de un sujeto sobre un objeto, un ejemplo es *SALIENTE* con respecto al objeto *LLAMADA*.
fuente: Es la fuente de donde se obtuvieron los datos, donde el atributo *numero* es el número de teléfono del dispositivo.

Entidad complementaria
contacto: Representa a los contactos del dispositivo investigado.

Vínculos
Relación circular múltiple de *registro_de_auditoria*, su objetivo es representar la dependencia de un registro de auditoría con sus predecesores, la que se origina cuando un registro de auditoría es generado como resultado de un análisis.

Relación múltiple entre *registro_de_auditoria* y *sujeto*, entre *registro_de_auditoria* y *objeto* y entre *registro_de_auditoria* y *accion*. La restricción referida a estas asociaciones es que el sujeto, objeto y acción se correspondan con la representación de "un sujeto realiza una acción sobre un objeto". La multiplicidad es consecuencia de que un registro de auditoría puede tener registros de auditoría como predecesores, dando lugar a que un registro de auditoría represente las acciones realizadas por sujetos sobre objetos.

[15] Las entidades corresponden a la adaptación del Modelo de Dorothy E. Denning al contexto del trabajo

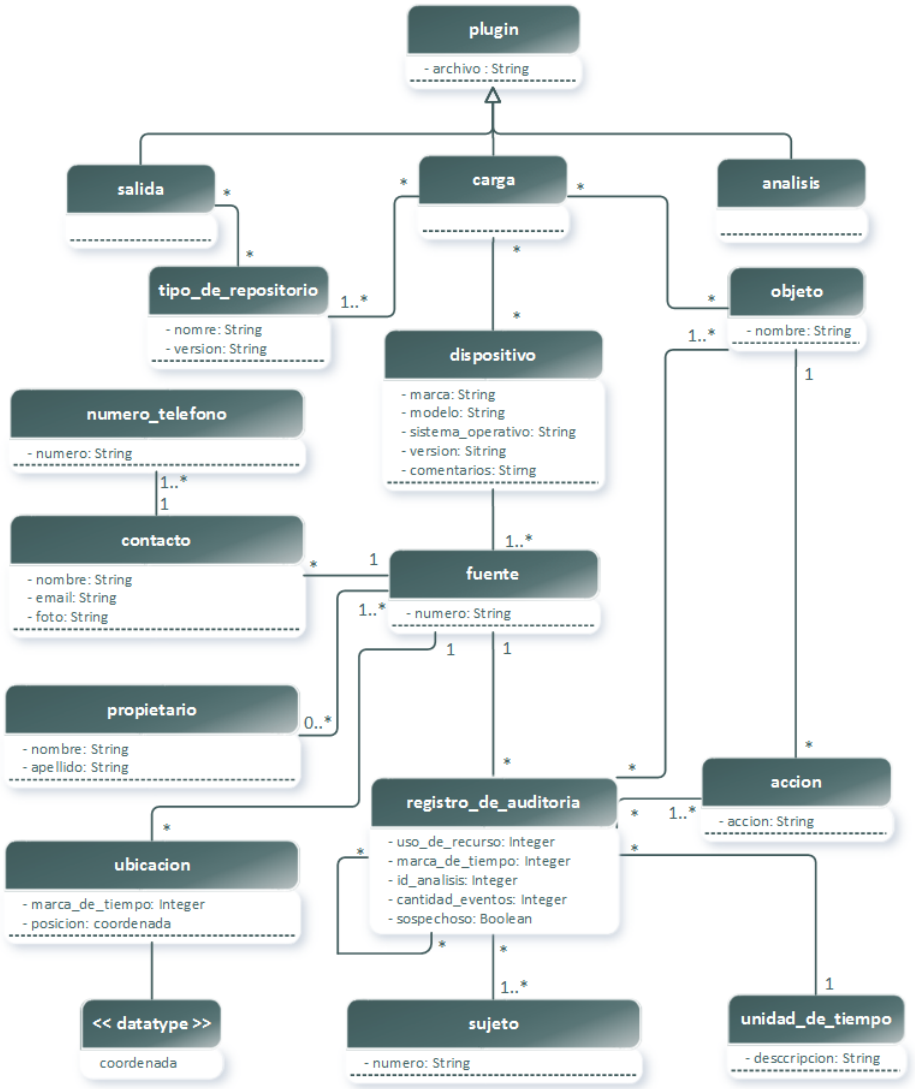

Figura 14: Modelo de dominio - Core

3.8.2. Extensión de análisis para detectar comportamiento sospechoso

Entidades asociadas al modelo de referencia

perfil: Representa el perfil, que se genera en base al template que se aplica a los registros de auditoría que se analizan.

template: Representa el template, donde se define la forma en la que serán considerados los registros de auditoría cuando estos sean analizados.

registro_de_anomalia: Representa al registro de anomalía y es donde se almacena el resultado de evaluar los registros de auditoría que configuran un comportamiento sospechoso.

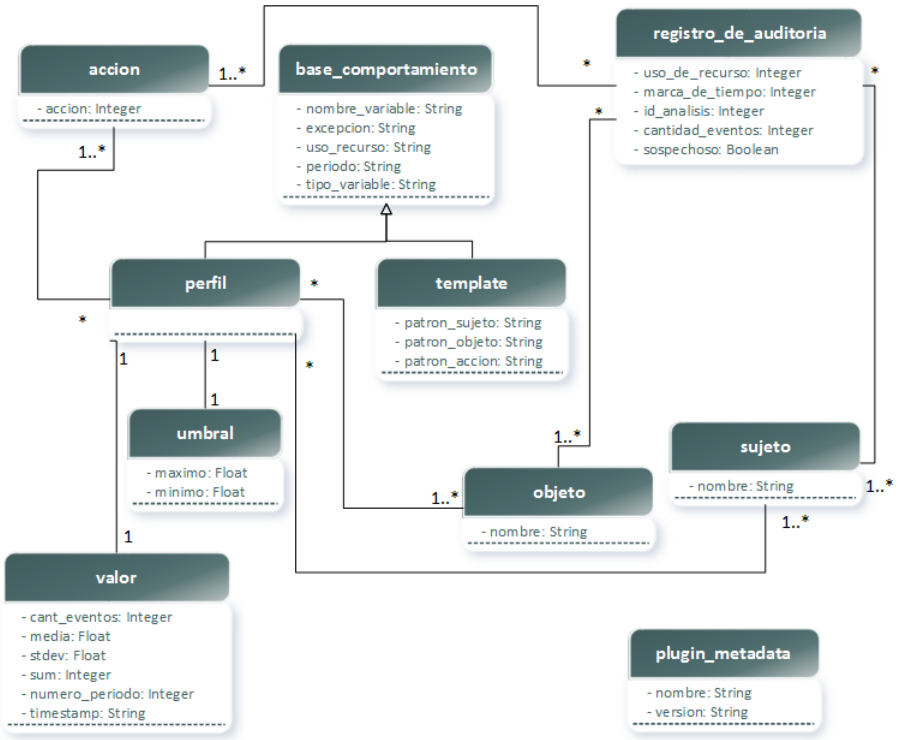

Figura 15: Modelo de dominio - Extensión de análisis para detectar comportamiento sospechoso

4. Diseño

4.1. Decisiones de diseño

Extensibilidad

La extensibilidad se la soporta a través del agregado de plugins, los mismos se utilizarán para la carga de datos, análisis de éstos y para desplegar la salida de la información. De esta forma se logra una herramienta que no depende del formato en el cual se brindan los datos de entrada, ofrece la capacidad de agregar distintos tipos de análisis y logra mostrar los resultados de los análisis de distintas formas.

Responsabilidades y funcionalidad

- La responsabilidad de analizar la información es completa de los plugins, la idea es tener un core reducido que sirva de nexo entre los datos cargados, los análisis que se realicen sobre estos y las salidas que se puedan generar de esos resultados.
- La responsabilidad del core es de brindar una interfaz para los plugins de carga, de análisis y de salida. Así como el almacenamiento de resultados parciales de los análisis para su posterior re-análisis.
- Los sujetos, objetos y acciones son provistos por los plugins de carga al momento de cargar los registros de auditoría. Es de responsabilidad del core agregarlos al sistema, sin controlar que existan bajo otro nombre.
- Es responsabilidad de los plugins brindar información suficiente en caso de manejar bases de datos específicas para la carga, de esta manera el investigador podrá elegir el que mejor se adecue a sus necesidades.
- Los plugins de análisis y salida proporcionarán la lista de operaciones que son capaces de realizar sobre los datos.
- Es de responsabilidad del investigador elegir el plugin adecuado para cargar los datos.

Sobre la Interfaz que brinda comunicación entre el Core y los Plugins

Para el diseño de la interfaz se consideraron dos posibles líneas a seguir, una donde las operaciones a implementar por los plugins estén explícitamente definidas en la interfaz y otra donde a través de una operación explícitamente definida en la interfaz se pueda ejecutar cualquier operación que un plugin sea capaz de proveer.

El segundo camino tiene como ventajas respecto al primero mayor escalabilidad y flexibilidad.

Escalabilidad en el sentido de que no está acotada la cantidad de operaciones que un plugin puede realizar, ya que en el caso de definir explícitamente las operaciones, por ejemplo, éstas podrían ser cargar, analizar y exportar. Flexibilidad en el sentido de que un plugin puede implementar varias operaciones de un mismo tipo (Carga, Análisis o Salida).

Dado lo anterior se opta por la segunda opción, o sea, la interfaz brindará una operación (llamada por ejemplo “ejecutar”) que le permita ejecutar al plugin cualquier operación que éste provea. Esta decisión implica que la interfaz deba proveer una operación que permita obtener del plugin la lista de operaciones que ofrece así como de qué tipo son (Carga, Análisis o Salida). Otra consideración que se toma es que los parámetros de entrada/salida de la operación “ejecutar” (sugerida anteriormente a modo de ejemplo) sea de tipo Set(Set(<Cualquier Tipo de Dato>)), flexibilizando así los tipos de datos que cada operación es capaz de manejar. Estando los tipos de datos factibles de usar acotados al conjunto de tipos de datos manejados por el core y definidos en la interfaz.

Cada operación podrá brindar un asistente para su uso o configuración, el objetivo de contar con el asistente de la operación es de dotar a ésta de un mecanismo que le permita tanto maximizar el beneficio de su uso como el de mejorar la experiencia del usuario.

Manejo de la marca de tiempo considerando el huso horario

Como se comentó en la etapa de análisis, al momento de analizar los eventos ocurridos es importante considerar el huso horario, por lo que se incluirá al momento de realizar el análisis la opción para especificar el huso horario de los datos a procesar.

4.2. Arquitectura

Partiendo de los requerimientos definidos no se infiere que sea necesario que la herramienta cuente con características como ser: aplicación distribuida, multi-usuario y tener alta disponibilidad.
Por otro lado si se considera necesario que cuente con la posibilidad de manejar grandes volúmenes de datos, distintos formatos de evidencia, capaz de exportar los resultados generados a otros formatos, sea escalable y poratable para ejecutarse en diferentes plataformas.

Considerando lo dicho anteriormente se decide utilizar el estilo arquitectónico basado en plugins. A continuación se presenta un esquema del mismo.

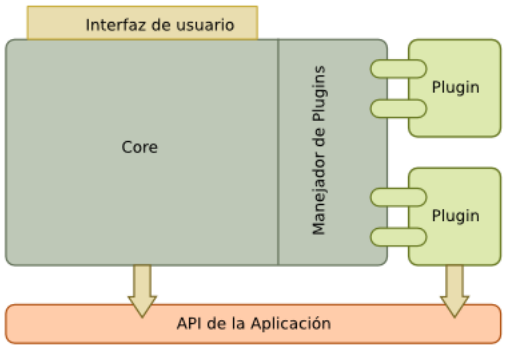

Figura 16: Estilo arquitectónico

A continuación se muestra un diagrama de componentes de la aplicación

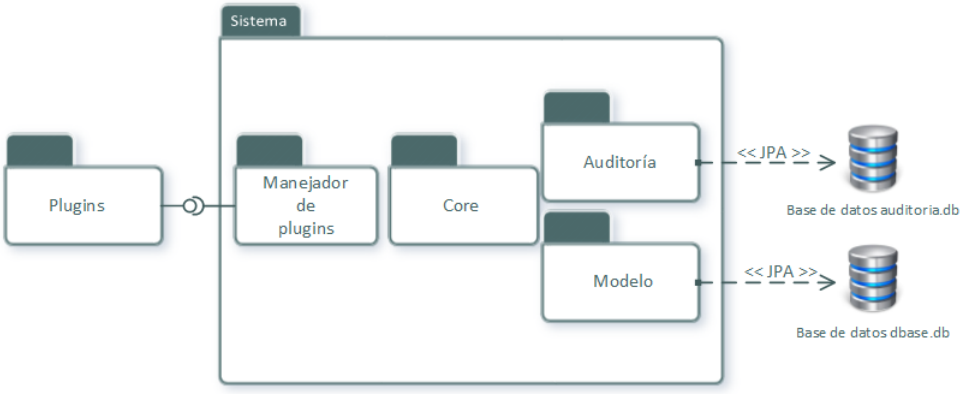

Figura 17: Modelo de la arquitectura

4.3. Interfaz para los Plugins

Los plugins que se desarrollen para interactuar con la herramienta deben estar empaquetados dentro de un archivo .jar y contener en su manifest.mf el tag *Plugin-Class:* seguido del nombre de la clase principal que implementa las operaciones de la interfaz.

Tipos de datos definidos

Estructura utilizada para representar la metadata que el sistema requiere de un plugin.

Atributo	**Tipo**	**Descripción**
nombre	String	Nombre del plugin
version	String	Versión del plugin
descripcion	String	Descripción general del plugin

Tabla 17: Tipo de dato DataMetaPlugin

Las operaciones implementadas por los plugins tienen que ser de alguno de los tipos definidos en la siguiente tabla.

Valor	**Descripción**
CARGA	Tipo asociado a las operaciones que permiten cargar datos al sistema
ANALISIS	Tipo asociado a las operaciones que permiten analizar los datos del sistema
SALIDA	Tipo asociado a las operaciones que permiten exportar los datos del sistema

Tabla 18: Enumerado tipo_operacion

Estructura utilizada para representar las operaciones que implementaran los plugins.

Atributo	**Tipo**	**Descripción**
nombre_plugin	String	Nombre del plugin
nombre_operacion	String	Nombre de la operación
id_operacion	Integer	Identificador de la operación
tipo	tipo_operacion	Tipo de la operación
descripcion_operacion	String	Descripción de la operación
parametros	String[ ]	Parámetros de la operación

Tabla 19: Tipo de dato operaciones_plugins

Estructura utilizada para representar los datos asociados a un contacto.

Atributo	**Tipo**	**Descripción**
nombre	String	Nombre del contacto
email	String	Dirección de correo electrónico
foto	String	Nombre de archivo y ubicación en el sistema de archivos de la foto
telefonos	List<String>	Lista de números telefónicos asociada al contacto
fuente	String	Fuente de la que se obtuvo el contacto

Tabla 20: Tipo de dato DataContacto

Estructura utilizada para representar un registro de auditoría, donde se representa la acción de un sujeto sobre un objeto.

Atributo	**Tipo**	**Descripción**
marca_de_tiempo	long	Marca de tiempo correspondiente al registro de auditoría, expresada en milisegundos como la cantidad de tiempo a partir del 1/1/1970 0:00:00 GMT
cantidad_eventos	int	Cantidad de eventos registrados
uso_de_recurso	int	Cantidad de tiempo en que se usa un recurso expresada en segundos
id_analisis	int	Identificación del análisis asociado al registro de auditoría
predecesores	List<DataRegistroAuditoria>	Lista de registros de auditoría que tras un proceso de análisis generaron este registro de auditoría
rosa	List<DataSOA>	Lista de tuplas (sujeto, objeto, accion) asociadas al registro de auditoría
valor_analisis	double	Valor asociado al registro de auditoría calculado por el proceso de análisis que generó dicho registro
limite_inferior	double	Valor asociado al registro de auditoría calculado por el proceso de análisis que generó dicho registro, es el menor valor que puede tomar *valor_analisis* para no ser considerado como sospechoso
limite_superior	double	Valor asociado al registro de auditoría calculado por el proceso de análisis que generó dicho registro, es el mayor valor que puede tomar *valor_analisis* para no ser considerado como sospechoso
sospechoso	boolean	Valor que indica si el registro de auditoría es considerado como sospechoso por el proceso de análisis que lo generó, es False si *limite_inferior* <= *valor_analisis* <= *limite_superior*, sino es True

Tabla 21: Tipo de dato DataRegistroAuditoria

Operaciones definidas

Los plugins que se creen para ser utilizados desde la herramienta desarrollada deben implementar las siguientes operaciones.

Método	**Entrada**	**Salida**
obtener_metadatos	—	DataMetaPlugin
obtener_operaciones	tipo_operacion...	List<operaciones_plugins>
ejecutar	int, String, Set<?>[16]	Set<?>
asistente	int, Set<?>	Set<?>

Tabla 22: Operaciones de la interfaz para plugins

El método *obtener_metadatos* es utilizado para obtener la información asociada al plugin, concretamente se obtiene su nombre, versión y descripción general.

El método *obtener_operaciones* es utilizado para obtener la lista de las operaciones que el plugin provee, opcionalmente recibe parámetros para indicar el tipo de operación específica que se desea obtener.

El método *ejecutar* es utilizado para ejecutar la operación que el plugin provee, la que está identificada con el valor de tipo int que recibe por parámetro. Además se pasan dos parámetros, uno es un String que en el caso de un plugin de CARGA o SALIDA se utiliza para indicar el nombre del archivo, con su ruta,

[16] Representa un conjunto de cualquier tipo de dato

que se leerá o se creará según el caso. El otro parámetro pasado es de tipo Set<?>, la utilización de este parámetro cobra relevancia en los plugins de SALIDA y ANALISIS, por ejemplo suponga que se desea exportar a un archivo de salida los registros de auditoría y los contactos cargados en el sistema, en ese caso el tipo de dato pasado sería Set< Set<DataRegistroAuditoria>, Set<DataContacto> >.
El resultado de la operación es retornado como un Set<?>. Si por ejemplo se realiza una carga de registros de SMSs enviados y recibidos el plugin de carga usado retornará a la herramienta un Set< Set<DataRegistroAuditoria> >.

El método *asistente* es utilizado para ejecutar el asistente de la operación que el plugin implementa, la que es identificada con el valor de tipo int que recibe por parámetro. Además se pasa por parámetro un dato de tipo Set<?>, que se usa de forma similar a lo expuesto en el método *ejecutar*.

4.3.1. Flujo de la carga de datos

En esta sección se muestra un flujo típico del proceso de carga de datos que ilustra el funcionamiento de los componentes del sistema.

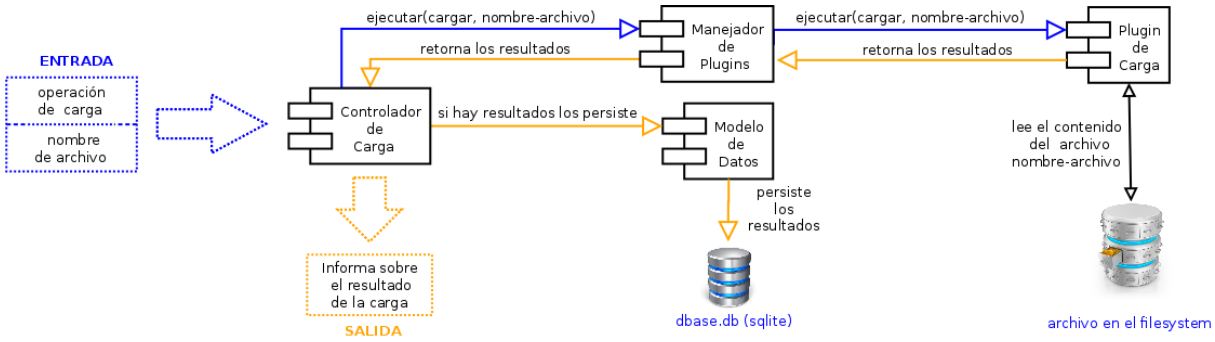

Figura 18: Flujo típico de carga de datos

A continuación se brinda una descripción más detallada del funcionamiento de cada componente y sus respectivas comunicaciones.

- **Controlador de carga:** Recibe de la capa de presentación la *operación de carga* y el path del archivo de donde consumir los datos. Esta información es entregada al **Manejador de plugins** para que ejecute dicha operación y retorne los datos a cargar. Los datos retornados por el **Manejador de plugins** son persistidos en el sistema.
- **Manejador de plugins:** A través de la *operación de carga* éste obtendrá el plugin y la operación del mismo que se deberá ejecutar. Luego instancia al plugin correspondiente y ejecuta la operación *ejecutar* con los parámetros operación y el path del archivo de donde leer. Cuando finaliza la ejecución del plugin, el **Manejador de plugins** devuelve los datos.
- **Plugin de carga:** Recibe cuál es la operación a ejecutar y el archivo a leer, éste leerá dicho archivo e interpretará los datos como corresponda, luego los retornará según los tipos de datos definidos en la interfaz **IPlugin**.

4.3.2. Flujo de la exportación de datos

En esta sección se muestra un flujo típico del proceso de exportación de datos que ilustra el funcionamiento de los componentes del sistema.

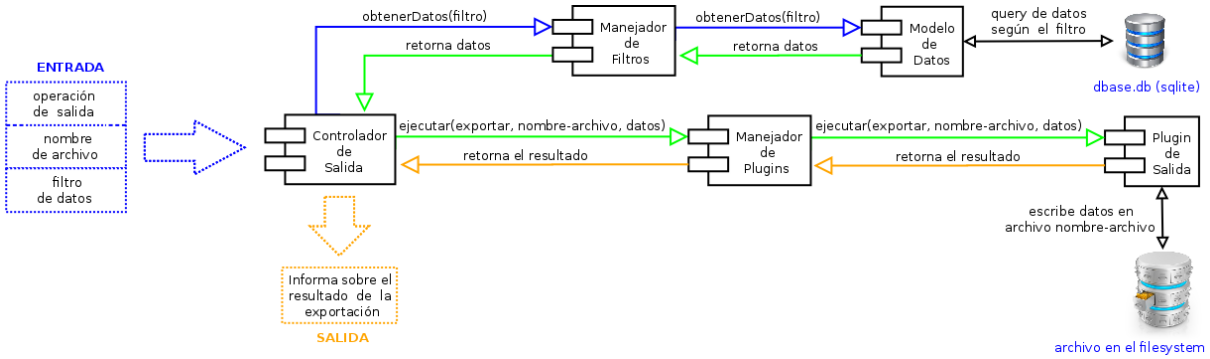

Figura 19: Flujo típico de exportación de datos

A continuación se brinda una descripción más detallada del funcionamiento de cada componente y sus respectivas comunicaciones.

- **Controlador de salida:** Recibe de la capa de presentación la *operación de salida*, el nombre del archivo a donde guardar los datos y el filtro aplicado. Le solicita los datos al **Manejador de filtros** brindándole el filtro aplicado y éste le devuelve el conjunto de datos a exportar. Luego habiendo obtenido los datos le encarga la exportación al **Manejador de pugins** brindándole la *operación de salida* y el nombre del archivo.

- **Manejador de plugins:** A través de la *operación de salida* éste obtendrá el plugin y la operación del mismo que se deberá ejecutar. Luego instancia al plugin correspondiente y ejecuta la operación *ejecutar* con los parámetros operación, el nombre del archivo a donde guardar y los datos a exportar.

- **Plugin de salida:** Recibe cuál es la operación a ejecutar, los datos a exportar y el nombre del archivo donde almacenar los datos exportados, éste interpretará los datos como corresponda y luego los exporta a un archivo CSV en el path indicado.

4.3.3. Flujo de análisis

En esta sección se muestra un flujo típico del proceso de análisis de datos que ilustra el funcionamiento de los componentes del sistema.

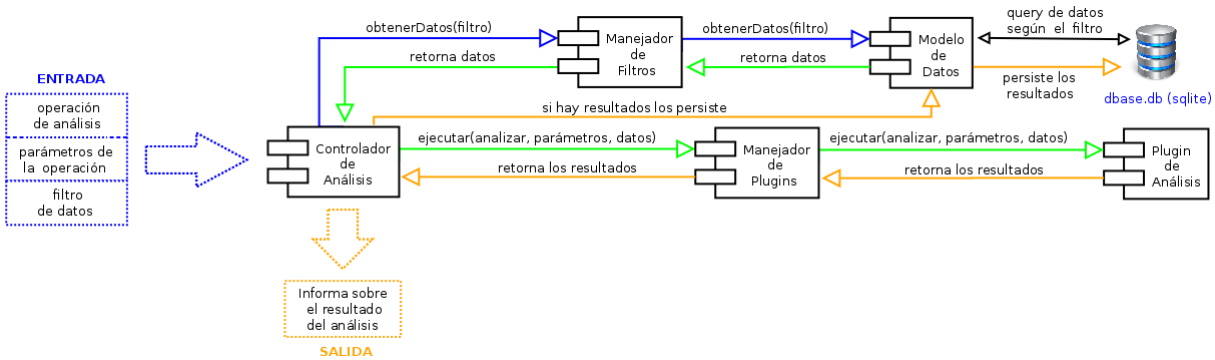

Figura 20: Flujo típico de análisis de datos

A continuación se brinda una descripción más detallada del funcionamiento de cada componente y sus respectivas comunicaciones.

- **Controlador de análisis:** Recibe de la capa de presentación la *operación de análisis*, los parámetros (en caso de que la operación los requiera) y el filtro aplicado. En caso de que la operación brinde un asistente para la configuración de los parámetros, los mismos podrán ser cargados a través de dicha interfaz. Acto seguido el **Controlador de análisis** le solicita los datos al **Manejador de filtros** brindándole el filtro aplicado y éste le devuelve el conjunto de datos a analizar. Luego, habiendo obtenido los datos, éste le encarga el análisis de los mismos al **Manejador de plugins** brindándole la *operación de análisis*, los datos y los parámetros. Cuando el **Manejador de plugins** termina el procesamiento devuelve los datos generados en el análisis los cuales se almacenan en el sistema. Por otra parte el **Controlador de análisis** genera una instancia de **Análisis** con los valores de filtro, operación de análisis y parámetros utilizados, permitiendo así su posterior revisión.
- **Manejador de plugins:** A través de la *operación de análisis* éste obtendrá el plugin y la operación del mismo que se deberá ejecutar. Luego instancia al plugin correspondiente y ejecuta la operación *ejecutar* con los parámetros operación, los datos a analizar y los parámetros.
- **Plugin de análisis:** Recibe cuál es la operación a ejecutar, los datos a analizar y un conjunto de parámetros, éste interpretará los datos como corresponda, los analizará y luego los retornará al sistema según se haya configurado.

4.3.4. Auditoría del sistema

En esta sección se expone el diseño de la auditoría del sistema.

Se le otorga la responsabilidad de proveer las operaciones para persistir la auditoría al Manejador de auditoría definido como *Singleton* para así tener acceso global desde los componentes del sistema.

Se observa que los lugares en donde se actualiza la base de datos son los procesos de carga y análisis de datos, por ende desde las operaciones que se encargan de estos procesos se deberá llamar a la operación global que el Manejador de auditoría publique.

A su vez se cree conveniente hacer un chequeo de la integridad de la base de datos del sistema al iniciar el mismo (para advertir en caso de que ésta haya sufrido cambios por fuera de la aplicación). Para ello lo que se debe hacer es registrar el hash de la base de datos (y el algoritmo utilizado) al cerrar la aplicación, cuando se inicia la misma se lo calcula y se lo compara con el último almacenado (correspondiente a un cierre de aplicación), si son distintos se advierte.

4.4. Modelo de datos del sistema

En esta sección se describen las estructuras de datos más relevantes.

4.4.1. RegistroDeAuditoría

Estructura utilizada para representar un registro de auditoría, en la cual se registra la ocurrencia de un evento, esto es, la acción de un sujeto sobre un objeto en un momento dado.

Atributo	Tipo	Descripción
marca_de_tiempo	Long	Marca de tiempo expresada en milisegundos como la cantidad de tiempo a partir del 1/1/1970 0:00:00 GMT
cantidad_eventos	Integer	Cantidad de eventos registrados
uso_de_recurso	Integer	Cantidad de tiempo en que se usa un recurso expresada en segundos
id_analisis	Integer	Identificación del análisis
predecesores	List<RegistroAuditoria>	Lista de registros de auditoría que tras un proceso de análisis generaron este RegistroDeAuditoría
rosa	List<ROSA>	Lista de tuplas (Sujeto, Objeto, Acción) asociadas a este RegistroDeAuditoría
valor_analisis	Double	Valor asociado al registro de auditoría calculado por el proceso de análisis que generó dicho registro
limite_inferior	Double	Valor asociado al registro de auditoría calculado por el proceso de análisis que generó dicho registro, es el menor valor que puede tomar *valor_analisis* para no ser considerado como sospechoso
limite_superior	Double	Valor asociado al registro de auditoría calculado por el proceso de análisis que generó dicho registro, es el mayor valor que puede tomar *valor_analisis* para no ser considerado como sospechoso
sospechoso	boolean	Valor que indica si el registro de auditoría es considerado como sospechoso por el proceso de análisis que lo generó, es False si *limite_inferior <= valor_analisis <= limite_superior*, sino es True

Tabla 23: Estuctura RegistroDeAuditoria

4.4.2. Sujeto

Estructura utilizada para representar un sujeto, en este sistema un sujeto hace referencia a quién realiza la acción.

Atributo	Tipo	Descripción
sujeto	String	Identificador del sujeto

Tabla 24: Estructura Sujeto

4.4.3. Objeto

Estructura utilizada para representar un objeto del sistema, en ésta se registra sobre cuál objeto se realiza la acción.

Atributo	Tipo	Descripción
nombre	String	Nombre del objeto sobre el cuál se realiza la acción

Tabla 25: Estructura Objeto

4.4.4. Accion

Estructura utilizada para representar una acción de un objeto.

Atributo	Tipo	Descripción
accion	String	Acción realizada sobre el objeto
objeto	Objeto	Objeto al que pertenece la acción

Tabla 26: Estructura Accion

4.4.5. Analisis

Estructura utilizada para representar un análisis, en la cual se registra el plugin utilizado, la operación, el filtro aplicado y la lista de parámetros usada con el fin de, en un momento posterior, inspeccionar cada análisis realizado.

Atributo	Tipo	Descripción
plugin	String	Nombre del plugin utilizado
operacion	String	Descripción de la operación utilizada
hash_plugin	String	Hash del plugin utilizado
algoritmo	String	Algoritmo utilizado para realizar el hash del plugin
parametros	List<ParametroPlugin>	Lista de parámetros utilizados en el análisis
filtro	Filtro	Filtro de datos utilizado para el análisis

Tabla 27: Estructura Analisis

4.4.6. Filtro

Estructura utilizada para representar un filtro, en la cual se registra el filtro utilizado al momento de realizar un análisis.

Atributo	Tipo	Descripción
fecha_desde	Calendar	Fecha inicio por la cual filtrar
fecha_hasta	Calendar	Fecha final por la cual filtrar
sujeto	String	Campo que representa al sujeto buscado
objeto	String	Campo que representa al objeto buscado
id_analisis	Integer	Identificador del análisis buscado
sospechoso	boolean	Indica si se desean buscar sólo los registros sospechosos o no
fuente	String	Identificador de la fuente de datos

Tabla 28: Estructura Filtro

4.4.7. ParametroPlugin

Estructura utilizada para representar un parámetro del plugin, en la cual se registra un campo y un valor que hacen referencia a la configuración del plugin en el momento de realizar el análisis.

Atributo	Tipo	Descripción
campo	String	Nombre del campo del parámetro
valor	String	Valor del parámetro referenciado en campo

Tabla 29: Estructura ParametroPlugin

4.4.8. ListaNormalizacion

Estructura utilizada para representar una lista de normalización, en la cual se registra la lista y su conjunto de reglas (patrones de matching y reemplazo).
El diseño de este módulo permite almacenar diferentes listas, las cuales podrán ser usadas dependiendo de la necesidad de la investigación en curso.

Atributo	Tipo	Descripción
nombre	String	Nombre de la lista de normalización
patrones	List<PatronesNormalizacion>	Conjunto de patrones que se aplicarán al usar la lista

Tabla 30: Estructura ListaNormalizacion

4.4.9. PatronesNormalizacion

Estructura utilizada para representar un patrón de normalización, en la cual se registra el patrón de matching y su reemplazo correspondiente.

Atributo	Tipo	Descripción
patron	String	Expresión regular que con la cual matchear los sujetos
reemplazo	String	Texto por el cual reemplazar aquellos sujetos que matcheen con el patrón

Tabla 31: Estructura PatronesNormalizacion

4.4.10. Contacto

Estructura utilizada para representar un contacto, en la cual se registran los contactos cargados.

Atributo	Tipo	Descripción
nombre	String	Nombre con el cual se agendó el contacto
email	String	Dirección de email
foto	String	Ruta del archivo que contiene la foto
telefonos	List<String>	Lista de números de teléfonos asociados al contacto
fuente	String	Identificador de la fuente de datos

Tabla 32: Estructura Contacto

4.5. Almacenamiento de datos

Para almacenar los datos de la aplicación se optaron por diferentes alternativas, archivo de configuración y bases de datos.

En el archivo de configuración, llamado *parametros.cfg* se definen las rutas de las bases de datos además del directorio donde almacenar los plugins.
Los parámetros soportados en dicho archivo son los siguientes:

- *plugins_dir*: Path del directorio donde se almacenarán los plugins.
- *auditoria_dir*: Path del directorio donde se almacenará la base de datos *auditoria.db*
- *base_dir*: Path del directorio donde se almacenará la base de datos *dbase.db*

En la base de datos *auditoria.db* se almacenarán todos los eventos realizados con la herramienta.
Un esquema de la misma se muestra a continuación:

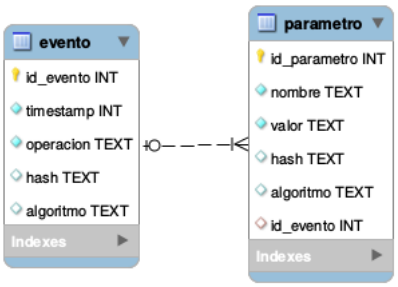

Figura 21: Modelo de la base de datos auditoria.db

En la base de datos *dbase.db* se almacenarán todos los datos cargados al sistema, así como también los resultados de los análisis e información adicional para poder soportar la lógica de la herramienta.
Un esquema de la misma se muestra a continuación:

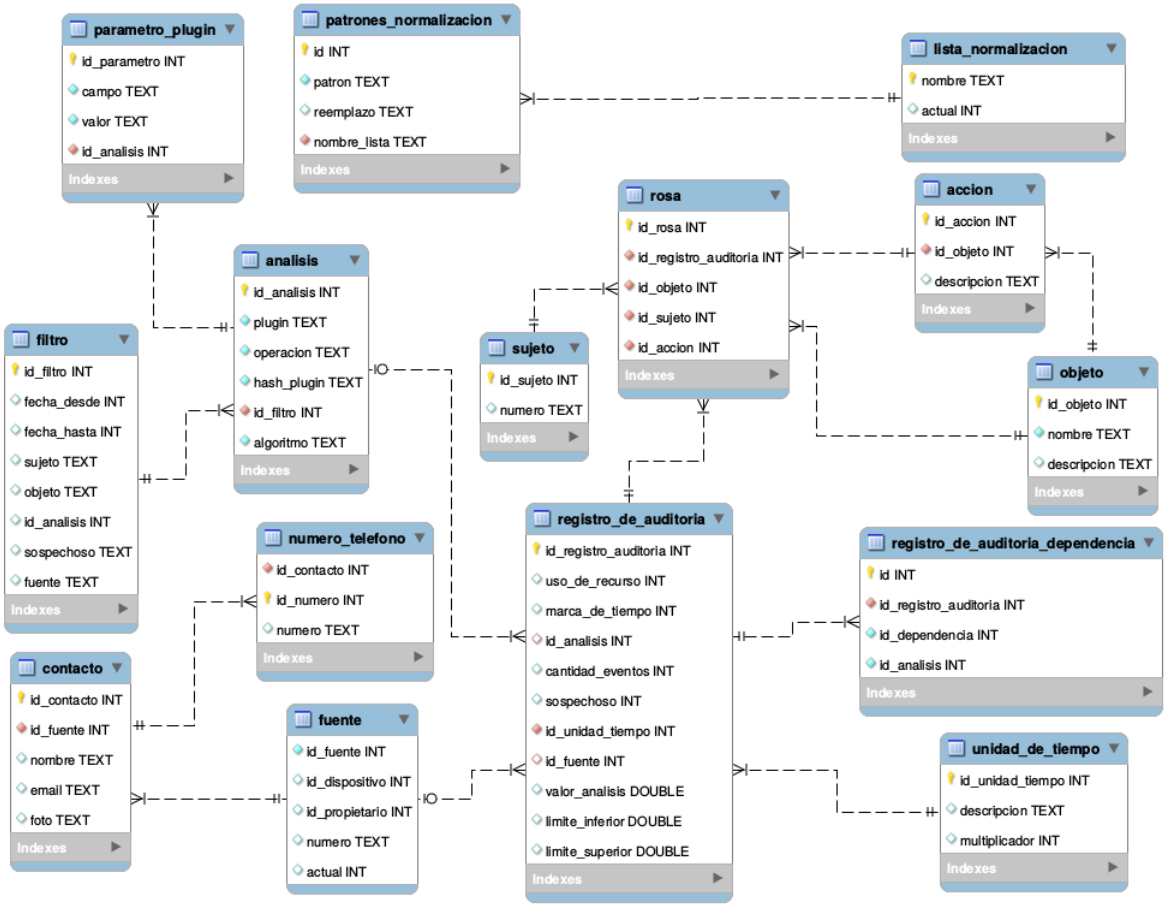

Figura 22: Modelo de la base de datos dbase.db

4.6. Diseño de un plugin de análisis

En esta sección se presentará uno de los puntos más importantes del trabajo realizado, el que hace al prototipo una herramienta que asiste al investigador forense en la tarea de identificar los datos relevantes para la investigación. En esa línea se diseña un plugin de análisis que basa su estrategia de identificación en la observación del comportamiento de sujetos sobre objetos que es caracterizado en términos de métricas y modelos estadísticos.

Concretamente en el plugin diseñado las métricas consideradas son *contador de eventos* y *medición de recursos* y los modelos estadísticos *Media y Desviación estándar* y *Multivariado*.

A continuación se describe el proceso realizado con un evento registrado, al que se llamará registro de auditoría, para a partir de sus datos y en combinación con eventos similares identificar un comportamiento que indique una actividad fuera de la habitual, la que conduce a que las primeras miradas de la investigación vayan sobre esos eventos, a los que se referirá como sospechosos.

La caracterización del comportamiento de sujetos respecto a objetos se realiza utilizando la estructura de perfil definida en la etapa de análisis. Como se recordará, éste es creado a partir de un template.
El template aporta la manera de agrupar registros de auditoría bajo un mismo perfil, por ejemplo agrupar por día las llamadas realizadas hacia un determinado número de teléfono.
Entonces como primer paso, a un registro de auditoría se le aplica un template, obteniendo como resultado un nuevo perfil.
Luego se busca ese perfil dentro de la colección de perfiles, si no existe se agrega a la colección de perfiles y se termina el proceso para ese registro.
Si existe, se determina si corresponden al mismo intervalo de tiempo, por ejemplo al mismo día, si es así los valores del nuevo perfil se acumulan en el existente y se finaliza el proceso para ese registro.
Si no coinciden, se analiza el perfil existente para determinar si se puede considerar sospechoso, si no es sospechoso se reemplaza el perfil existente por el nuevo y se finaliza el proceso para ese registro.
Si es sospechoso, se genera un registro de anomalía en base al perfil existente y se lo agrega a la colección de registros de anomalías, además se reemplaza el perfil existente por el nuevo y se finaliza el proceso para ese registro.

A modo de complementar la idea del proceso descripto se presenta el siguiente flujo.

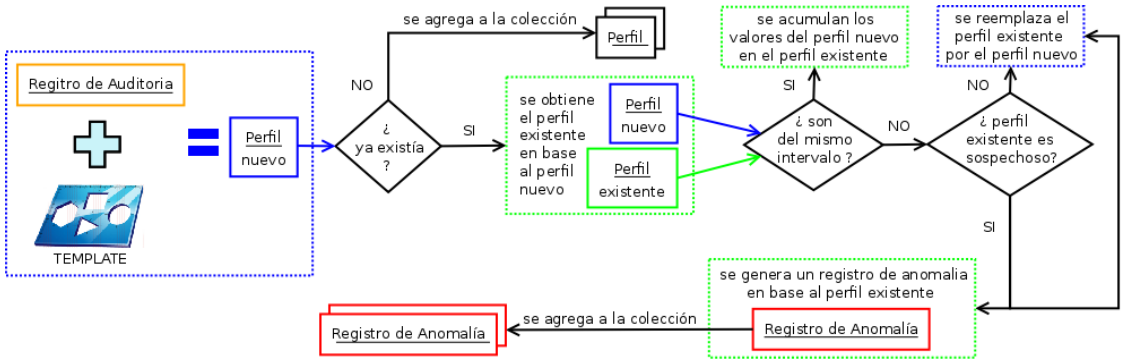

Figura 23: Flujo típico de análisis de un registro de auditoría

El proceso anteriormente descripto se repite para todos los registros de auditoría.
Antes de dar por culminado el análisis se determina si los perfiles existentes pueden considerarse sospe-

chosos, el que lo sea dará lugar a la generación de su respectivo registro de anomalía, que será agregado a la colección de registros de anomalía.
Con esto, el análisis queda terminado y se retorna como resultado los registros de anomalía generados.

Las partes del proceso interesantes a destacar son la generación del perfil y las que tienen toma de decisión, a saber, ¿ya existía?, ¿son del mismo intervalo? y ¿perfil existente es sospechoso?.

¿ya existía?
Lo que identifica a un perfil en este contexto son las tuplas *(sujeto, objeto, acción)* que lo componen, por ejemplo un perfil puede tener las siguientes tuplas: *(+59812345678, LLAMADA, ENTRANTE)* y *(+59812345678, LLAMADA, SALIENTE)*. Por lo tanto se dice que un nuevo perfil ya existía si sus tuplas *(sujeto, objeto, acción)* están incluidas en las de alguno de los perfiles existentes.

¿son del mismo intervalo?
El perfil tiene asociada una marca de tiempo, que se corresponde con la del registro de auditoría que lo generó, esa marca de tiempo cae dentro de un intervalo, por ejemplo un día específico, por lo que dos perfiles son del mismo intervalo si sus marcas de tiempo, volviendo al ejemplo, corresponden al mismo día.
El discernir esto permite decidir entre acumular los valores del nuevo perfil en el existente o dar lugar a un nuevo intervalo, en este último caso y antes de generar el nuevo intervalo se debe determinar si en el intervalo actual hay valores que indiquen un comportamiento fuera de lo habitual. Una consideración importante que se debe tener es que los registros de auditoría están cronológicamente ordenados antes de ser analizados.

¿perfil existente es sospechoso?
Un perfil se considera sospechoso si los valores que se acumulan, determinados por la métrica que se aplica, caen fuera del intervalo de confianza determinado por la aplicación del modelo estadístico a los valores asociados a los intervalos previos.
Por ejemplo, usando la métrica cantidad de eventos, el modelo estadístico media y desviación estándar y considerando los registros de llamadas del mes de Diciembre de 2014, si para el día 24 se registraron 15 llamadas y en los anteriores se tenían 2 llamadas por día, el intervalo de confianza [17] queda determinado entre 0,29 y 3,71, por lo que el perfil para este caso es señalado como sospechoso.

Generación del perfil
Un nuevo perfil es el resultado de aplicarle una transformación a un registro de auditoría a través de un template.
Por ejemplo, si se quiere estudiar el comportamiento asociado a las llamadas realizadas y recibidas con el número de teléfono +59812345678, tomando la cantidad de llamadas como métrica y un día como intervalo de acumulación de eventos, el template correspondiente sería:

patron-sujeto	**patron-objeto**	**patron-accion**	**métrica**	**intervalo**
Analizar el sujeto +59812345678	considerar solo los objetos LLAMADA	considerar solo las acciones ENTRANTE y SALIENTE	# eventos	86400000 [18] (milisegundos)

Tabla 33: Versión en alto nivel de los atributos del template relevantes al ejemplo

[17] Intervalo de confianza $= 2 \pm 4 \times \sqrt{(\frac{92}{22} - 4)}$
[18] 86400000 milisegundos = 1 día

La definición del template, las primeras veces, puede resultar una tarea difícil de realizar por lo que el plugin provee un asistente que facilita su construcción. Solo a modo ilustrativo se presenta la forma real que tendría el template del ejemplo, los detalles del lenguaje diseñado para describir los patrones del template se presentan en el anexo.

patron-sujeto	**patron-objeto**	**patron-accion**	**métrica**	**intervalo**
+59812345678 <- *	LLAMADA <- *	ENTRANTE|SALIENTE <- *	eventos	86400000

Tabla 34: Atributos del template relevantes al ejemplo

5. Implementación

5.1. Lenguaje

Requerimientos de la herramienta

La funcionalidad principal de la herramienta es ser un asistente para el análisis de evidencia, para esto debería poseer características gráficas con el fin de ayudar en la visualización de los datos. Además es conveniente que sea multiplataforma y deberá proveer la forma de añadir plugins.
Asimismo, debe proveer facilidades para mejorar la interfaz gráfica como por ejemplo una interfaz web.

Características no requeridas de la herramienta

- Acceso a datos en bajo nivel, es decir, no se requieren leer cabezales de archivos, ver el tráfico de la red, etcétera.
- Ejecución en tiempo real, ni alta performance.

Características deseables del lenguaje con el que se implementará la herramienta

- Sea orientado a objetos.
- Haya variedad de APIs disponibles.
- Esté bien documentado.
- La comunidad de desarrolladores en dicho lenguaje sea grande.

Dicho lo anterior, se piensa en lenguajes los cuales se acoplen mejor a estas necesidades y características deseables.
Algunos lenguajes que caen dentro de lo expresado son Python y Java, de los cuales se decide utilizar Java dada la experiencia previa.

5.2. Base de datos

Se opta por utilizar como base de datos SQLite ya que no requiere contar con un servidor de bases de datos instalado y tampoco requiere licencia debido a que es libre.

5.3. Entorno de desarrollo y ejecución

A continuación se detallan las aplicaciones y/o módulos con sus versiones utilizados en el desarrollo y la ejecución de la herramienta.

- Java JDK (Java Development Kit), versión 1.7
- IDE NetBeans, versión 8.0.2
- SQLite, versión 3
- EclipseLink (JPA 2.1)
- commons-io-2.4.jar
- commons-lang-2.6.jar
- commons-logging-1.2.jar
- jaxb2-basics-runtime-0.9.5.jar
- spring-core-4.2.2.RELEASE.jar
- STIX-1.2.0.2-SNAPSHOT.jar
- Sistemas operativos Linux Fedora core 19, Fedora core 20 y Ubuntu 14.04

5.4. Manejador de plugins

Para esta parte se utilizó como librería parte del código brindado por Roman Vottner en [34] el cual fue levemente modificado por el grupo para adaptarlo al contexto del proyecto. Éste código carga cada plugin en un ClassLoader diferente brindando así la posibilidad de cargar, descargar y actualizar plugins en tiempo de ejecución sin necesidad de cerrar la aplicación.

Entender el funcionamiento de partes claves del código fue en si mismo un trabajo complicado ya que utiliza muchas utilidades del lenguaje que eran nuevas para el grupo, en este documento no se entrará en una descripción detallada del mismo ya que escapa del objetivo principal de este proyecto y se encuentra debidamente documentado en el código de la referencia indicada.

A continuación se hará una breve de descripción del componente *PluginFramework* que fue el que se utilizó en la herramienta.

PluginFramework

Cuenta con clases para manejar en alto nivel los plugins como cargar, descargar, recargar y obtener las instancias de los plugins a través de su nombre. Además cuenta con el componente *PluginCore* que provee dos funcionalidades, la primera es la de detectar cambios en el directorio donde se encuentran los plugins (*filemonitor*) y la segunda funcionalidad es la de cargar/descargar los plugins más a bajo nivel (*classloader*).

filemonitor: La técnica que utiliza para detectar cambios es la realización de polling sobre el directorio para luego comparar los archivos ahí encontrados con los que tiene registrados, si falta uno notifica para que se dé de baja, si encuentra coincidencia entre lo que tiene registrado y los archivos actuales chequea la fecha de modificación, si coinciden lo descarta y si no notifica que el archivo fue modificado, por último, si encuentra plugins nuevos notifica la existencia de los mismos.

classloader: En éste componente es donde se encuentra el código del cargador de las clases que implementan la interfaz de los plugins, redefine el class loader del sistema. Básicamente lo que hace es buscar la clase que implementa la interfaz dentro del archivo .jar y lo carga en un ClassLoader separado a los demás plugins.

Ejemplo

A continuación se muestra un ejemplos de uso de los plugins a través del Manejador de plugins.

Operación *ejecutar* que implementa la interfaz del Manejador de plugins en el controlador.

```
@Override
public Set<?> ejecutar(String plugin, int id_operacion, String path, Set<?> datos) {
   // Se instancia al manejador
    PluginManager manager = SimplePluginManager.getInstance();
    // Se instancia al plugin que lleva por nombre 'plugin'
    IPlugin p = manager.getNewPluginInstance(plugin);
    // Se le indica al plugin que ejecute la operacion que se identifica con el ID
        'id_operacion'
    return p.ejecutar(id_operacion, path, datos);
}
```

5.5. Implementación de plugins

Como se comentó en el capítulo de Diseño, el plugin que se desarrolle además de implementar las operaciones de la interfaz de Plugins debe tener un archivo de manifiesto donde indique cual es la clase que implementa la interfaz a través del TAG *Plugin-Class*.

Por ejemplo, si se crea un plugin de carga donde la clase que implementa la interfaz se llama PluginCarga organizada bajo el paquete plugincarga, dentro del .jar generado tras compilar el código debe existir el archivo META-INF/MANIFEST.MF y contener en una línea el valor:

Plugin-Class: plugincarga.PluginCarga

Otro punto interesante a destacar es el procedimiento a seguir si se desea que el archivo .jar generado contenga todas las dependencias del plugin. Dicho procedimiento consiste en agregar en el archivo build.xml (del Project NetBeans) las siguientes líneas:

```
<target name="package-for-store" depends="jar">

        <property name="store.jar.name" value="CybOX"/>
        <property name="store.dir" value="store"/>
        <property name="store.jar" value="${store.dir}/${store.jar.name}.jar"/>

        <echo message="Packaging ${application.title} into a single JAR at ${store.jar}"/>

        <delete dir="${store.dir}"/>
        <mkdir dir="${store.dir}"/>

        <jar destfile="${store.dir}/temp_final.jar" filesetmanifest="skip">
            <zipgroupfileset dir="dist" includes="*.jar"/>
            <zipgroupfileset dir="dist/lib" includes="*.jar"/>

            <manifest>
                <attribute name="Plugin-Class" value="${main.class}"/>
            </manifest>
        </jar>

        <zip destfile="${store.jar}">
            <zipfileset src="${store.dir}/temp_final.jar"
            excludes="META-INF/*.SF, META-INF/*.DSA, META-INF/*.RSA"/>
        </zip>

        <delete file="${store.dir}/temp_final.jar"/>

    </target>
```

Al momento de generar el .jar se debe seleccionar el build “package-for-store”.

5.5.1. Plugin de Análisis

A continuación se presentan partes de la implementación que se consideran importantes a destacar.

Normalización de la marca de tiempo

La corrección de la marca de tiempo, antes de utilizarla para hacer cálculos, es necesaria para evitar el problema descripto en la sub-sección "Importancia del huso horario" del capítulo Análisis.

La solución consiste en desplazar la marca de tiempo desde GMT 0 hasta la zona horaria asociada al lugar en donde se registró el evento.
Luego de esta corrección se pueden efectuar los cálculos que se deseen, en el caso del análisis que se realiza en este trabajo el cálculo que se hace es la determinación del número de intervalo al que corresponde el evento.
A continuación, se presenta un ejemplo para ilustrar la aplicación de la idea expuesta.

Datos de Entrada

```
marcaDeTiempo(evento 1) = 28/08/2014 10:40:29 de Uruguay
marcaDeTiempo(evento 2) = 28/08/2014 21:40:29 de Uruguay
longitudDelIntervalo = 1 día
zonaHoraria(evento 1) = zonaHoraria(evento 2) = America/Montevideo
```

Cálculos previos

```
t1 = milisegundos(marcaDeTiempo(evento 1)) = 1409233229000
t2 = milisegundos(marcaDeTiempo(evento 2)) = 1409272829000
l = milisegundos(longitudDelIntervalo) = 86400000
TZoffset = milisegundos(TimeZoneOffset(America/Montevideo)) = -10800000
```

Cálculo del número de intervalo

Determinación del intervalo sin normalizar la marca de tiempo

```
n1 = t1 div l = 16310
n2 = t1 div l = 16311
```

Determinación del intervalo normalizando la marca de tiempo

```
n1 = ( t1 + TZoffset ) div l = 16310
n2 = ( t2 + TZoffset ) div l = 16310
```

Nota: div corresponde a la división entera.

Como se puede apreciar, cuando no se normalizan las marcas de tiempo se puede dar que a marcas del mismo día les correspondan números de intervalos diferentes, lo cual está mal.

Determinación de comportamiento sospechoso

A continuación se muestra un pseudo-código correspondiente a la implementación de la parte del flujo que está entre "*¿ya existía?*" y "*se genera un registro de anomalía en base al perfil existente*" mostrado en la figura 23 : *Flujo típico de análisis de un registro de auditoría.*

```
nuevo_perfil = new Perfil(<parametros para crear el perfil>);
perfil_existente = obtener_perfil(perfiles, nuevo_perfil);

if (perfil_existente != null) {
   n1 = nro_intervalo(nuevo_perfil);
   n2 = nro_intervalo(perfil_existente);

   if (n1 == n2)
      perfil_existente.acumular(nuevo_perfil);

   else {
      if (perfil_es_sospechoso(perfil_existente)) {
         nuevo_reg_anomalia = new RegistroAnomalia(perfil_existente);
         // El nuevo registro de anomalia se agrega a la coleccion de registros de anomalia
         registros_anomalia.add(nuevo_reg_anomalia);
      }
      // El perfil existente se reemplaza en la coleccion de perfiles por el nuevo
      perfil_existente.nuevo_intervalo(nuevo_perfil);
   }
}
else
   // El nuevo perfil se agrega a la coleccion de perfiles
   perfiles.add(nuevo_perfil);
```

5.6. Vinculación entre Registros de auditoría y Contactos

Algo que a priori puede sonar tan trivial como *obtener las comunicaciones de un contacto* no lo es pensando en el diseño orientado al modelo propuesto por Dorothy E. Denning en el que está basado la herramienta, esto se debe a que el concepto Contactos no existe en dicho modelo y por ende es necesario implementar ese vínculo.

Como primer punto a destacar se tiene que la herramienta no agrupa contactos, es decir, no toma decisiones al encontrarse con Contactos repetidos, por el hecho de que más de un Contacto pueden tener el mismo nombre, una misma persona puede estar agendada de diferentes formas, etcétera. Se propone esta tarea como trabajo futuro.
Introducido lo anterior, se pueden dar casos como los que se muestran en la figura 24 donde se ve que *Patrick Payge* aparece dos veces con dos números de teléfonos distintos.

Nombre	Teléfono
PATRICK PAYGE	+79099636378
PATRICK PAYGE	+79688691663

Figura 24: Contactos con igual nombre *PATRICK PAYGE* y diferentes números de teléfono

Repasando, un Registro de auditoría cuenta con un campo ***sujeto***, el mismo hace referencia a ***quién*** realizó una acción sobre un objeto, por ende es el *sujeto* el que se va a vincular con los Contactos, las preguntas que surgen son ¿cómo? y ¿a través de qué?, para responder a las cuestiones anteriores vale hacer la siguiente aclaración: en el campo *sujeto* de un Registro de auditoría (para la herramienta desarrollada y con los plugins provistos) puede contener tanto un número de teléfono o bien una dirección e-mail. Dicho lo anterior se tiene la clave del vínculo entre los Contactos y los Registros de auditoría.

En resumen, para un Contacto dado se debe buscar si su dirección de e-mail o alguno de sus números de teléfonos asociados a él coinciden con el campo *sujeto* de los Registros de auditoría, cada ocurrencia sería considerada como una comunicación realizada con el Contacto.

6. Caso de estudio

En esta sección se mostrará, en el contexto de la etapa de análisis de una investigación forense llevada adelante por un investigador, el uso de la herramienta desarrollada (la que identificaremos como AsInFoD) con el objetivo de mostrar como la misma reduce el conjunto de eventos por los que iniciar la investigación, por medio de la determinación de los que presentan mayor probabilidad de configurar un comportamiento fuera de lo habitual.

De los conjuntos de datos disponibles se usará uno de los proporcionados por Oxygen Forensics, ya que los mismos tienen como objetivo simular casos reales para evaluar el funcionamiento de su suite de software forense, lo que en cierta medida permite ver como responde la herramienta cuando es sometida a prueba, a priori, en un escenario real de un proceso de investigación.
Por ejemplo, se usará el *backup* correspondiente a un smartphone *HTC EVO 3D* modelo *X515m* con *Android* version *2.3* [35].

Luego de tener el conjunto de datos a analizar se presenta el primer desafío, y es, ¿AsInFoD es capaz de interpretar el formato en el que están esos datos?
La respuesta es NO, entonces ¿Qué se puede hacer?.
Para responder esa pregunta, primero se muestran en la tabla 35 los datos de entrada soportados por AsInFoD.

Tipo de dato	**Formato**	**Comentario**
SMS	CybOX	Compatible con la definición del objeto SMSMessage de la version 2.1 de CybOX
eMails	CybOX	Compatible con la definición del objeto EmailMessage de la version 2.1 de CybOX
Contactos	CybOX	Compatible con la definición del objeto Contact [2] para la version 2.1 de CybOX
SMS	sqlite	Compatible con Android (2.3.6) utilizado por Samsung Galaxy Ace 2 (I8160-L)
Llamadas	sqlite	Compatible con Android (2.3.6) utilizado por Samsung Galaxy Ace 2 (I8160-L)
Contactos	sqlite	Compatible con Android (2.3.6) utilizado por Samsung Galaxy Ace 2 (I8160-L)

Tabla 35: Datos de entrada soportados por AsInFoD

Dado que no es el objetivo central de la herramienta, la misma no soporta todo el universo de posibles representaciones de datos, pero como se indica en la tabla cuenta con la posibilidad de interpretar el formato del lenguaje CybOX, que también es soportado por la herramienta *Android Inspector* desarrollada por el proyecto de grado *"Recolección de evidencia digital sobre dispositivos móviles"* [2], que permite extraer los datos de los dispositivos móviles a los que representa en un formato estándar (CybOX) y los deja disponibles para consumir por otras herramientas.
Entonces, volviendo a la pregunta planteada en el marco del desafío, la solución es utilizar *Android Inspector* para obtener una representación de esos datos en formato CybOX.

Siguiendo con el caso de estudio, el mismo se desarrollará mostrando una posible secuencia de pasos realizada por un investigador, para la que se define la siguiente hipótesis: "el investigador no tiene más información que los propios datos que fueron recolectados".

Uso de la herramienta

Luego de ejecutar la herramienta, el investigador define la fuente a la que corresponden los datos que en el siguiente paso se cargarán, para este caso se definirá "Oxygen".
Para cargar los datos elige los archivos CybOX que contienen los eventos a analizar y las operaciones correspondientes ofrecidas por el plugin que se encarga de leer datos CybOX. Al llevar a cabo el proceso se pueden observar que se cargan 63 eventos (Registros de auditoría) en el sistema.

Figura 25: Resultado de la carga de SMSs desde un origen CybOX

Con los datos ya cargados en el sistema, el investigador procede a realizar un análisis de los mismos con el fin de observar en qué días el dispositivo investigado presentó una actividad sospechosa, para ello se elije dentro de los análisis disponibles la operación que permite encontrar actividad sospechosa a través del análisis estadístico utilizando como métrica *contador de eventos* [19]. Para ello se configuran los parámetros del plugin de análisis de la siguiente manera,

- Considerar todos los sujetos pero NO discriminarlos.
- Considerar todos los objetos pero NO discriminarlos.
- Considerar todas las acciones pero NO discriminarlas.
- No mostrar en los resultados registros que no son sospechosos.
- Agrupar los eventos por día.

Luego que el mismo culmina se pueden ver solamente tres días con actividad sospechosa, reduciendo en el entorno del 95 % el volumen de eventos a revisar en esta primera etapa del análisis.

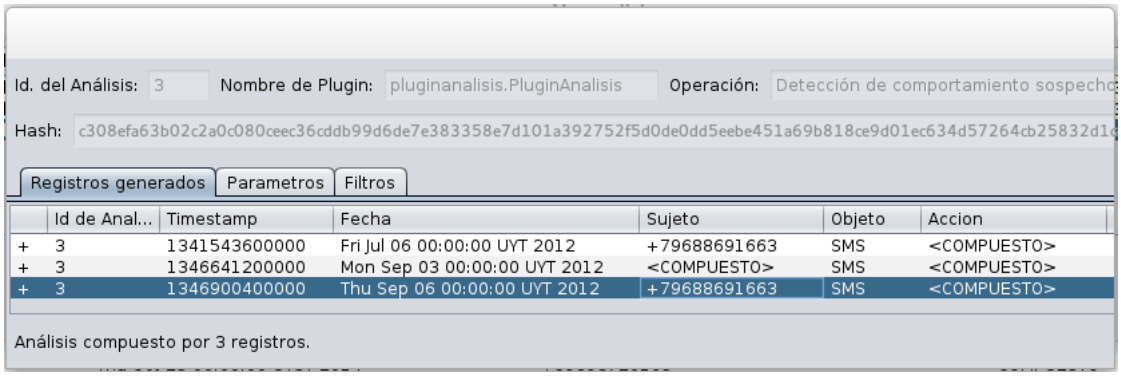

Figura 26: Resultado del análisis donde se muestran tres días con actividad sospechosa

Utilizando la utilidad "Ver análisis" se puede ver que en dos de los tres días las comunicaciones se realizaron con el mismo número (ver Figura 26), de aquí se destaca claramente un número donde focalizar la atención, *+79688691663* . Si además se desglosa el resultado del análisis el cual dice *"<COMPUESTO>"* en el campo sujeto se puede destacar otro número de teléfono con una cantidad de actividad interesante, este es *+79099636378* (ver Figura 27).

[19] El resto de las operaciones de análisis se las describirán en el Anexo

	idAnalisis	Timestamp	Fecha	Sujeto	Objeto	Accion	...
+	3	1346674498000	Mon Sep 03 09:14:58 UYT 2012	+79688691663	SMS	SALIENTE	...
+	3	1346674622000	Mon Sep 03 09:17:02 UYT 2012	+79688691663	SMS	ENTRANTE	...
+	3	1346675016000	Mon Sep 03 09:23:36 UYT 2012	0674	SMS	ENTRANTE	...
+	3	1346675838000	Mon Sep 03 09:37:18 UYT 2012	+79099636378	SMS	SALIENTE	...
+	3	1346675866000	Mon Sep 03 09:37:46 UYT 2012	+79099636378	SMS	ENTRANTE	...
+	3	1346675909000	Mon Sep 03 09:38:29 UYT 2012	+79099636378	SMS	SALIENTE	...
+	3	1346675938000	Mon Sep 03 09:38:58 UYT 2012	+79099636378	SMS	ENTRANTE	...
+	3	1346675970000	Mon Sep 03 09:39:30 UYT 2012	+79099636378	SMS	SALIENTE	...
+	3	1346675986000	Mon Sep 03 09:39:46 UYT 2012	+79099636378	SMS	ENTRANTE	...
+	3	1346676001000	Mon Sep 03 09:40:01 UYT 2012	+79099636378	SMS	SALIENTE	...
+	3	1346676312000	Mon Sep 03 09:45:12 UYT 2012	+79688691663	SMS	SALIENTE	...
+	3	1346681409000	Mon Sep 03 11:10:09 UYT 2012	+79688691663	SMS	ENTRANTE	...

Figura 27: Desglose del resultado del análisis donde se muestra cuales eventos lo generaron

Partiendo de los dos números de teléfonos destacados en la etapa anterior se utiliza el filtro por contactos[20] para tratar de vincular los números a nombres (y de ser posible con alguna foto). Se observa que para la búsqueda por el número *+79688691663* se devuelve el nombre "PATRICK PAYGE". Por otra parte, cuando se buscan los contactos con el número de teléfono *+79099636378* se pueden ver dos nombres distintos "PATRICK PAYGE" y "JOHN CROW". Para el caso de "JOHN CROW" se puede observar que tiene una foto en la agenda del teléfono, cuya ubicación es:

"file:///data/data/com.android.providers.contacts/files/thumbnail_photo_205.jpg"

Yendo a la ruta expuesta dentro de los datos investigados se observa la siguiente foto

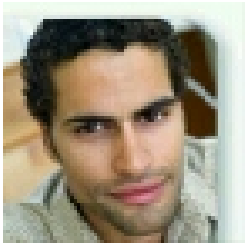

Figura 28: Foto del contacto llamado "JOHN CROW"

Expuesto lo anterior, queda en evidencia la reducción del volumen de información lograda a través del análisis del comportamiento del usuario del dispositivo móvil facilitando así el abordaje de la investigación en una primera instancia.

Análisis basado en un modelo estadístico: Los resultados esperados corresponden a un comportamiento fuera de lo habitual, ya sea en cantidad de eventos o en la duración de los mismos, ahora, un comportamiento fuera de lo habitual no necesariamente es la confirmación de que ocurrió un hecho delictivo, el discernir si corresponde o no a un delito es responsabilidad del investigador. En la descripción del análisis que se realiza para el caso se catalogará como falso positivo cuando la herramienta detecta comportamiento sospechoso y del contenido de la comunicación no puede afirmarse que tenga relación con una actividad delictiva, por otro lado se catalogará como falso negativo cuando la herramienta no indique focalizarse sobre el contenido de cierta comunicación y ésta si presente indicios de una actividad delictiva.

[20] Los cuales fueron cargados de forma similar a los eventos a analizar

Ahora, yendo a la fuente de los datos, se pueden obtener los contenidos de los SMS. Observando el intercambio de mensajes de los días marcados con actividad sospechosa:

Fecha 6/7/2012

Fecha	**Sentido (E/S)**	**Cuerpo del mensaje**
2012-07-06T03:57:34	S	Please call me asap
2012-07-06T03:58:05	E	I am on a business meeting.a bit later
2012-07-06T05:31:22	E	Thanks for the call
2012-07-06T07:51:57	S	Sorry the baby is sleeping.what's up*
2012-07-06T07:52:57	E	I just wanted to make sure that I am sending the papers to the right address
2012-07-06T07:54:01	S	**River road 45**
2012-07-06T07:54:30	E	Thanks

Tabla 36: Intercambios de mensajes del 06/07/2012

En este intercambio no se observa que la conversación sea en un contexto delictivo, con lo cual se puede catalogar como un falso positivo. Pero por otro lado se obtiene una dirección que puede ser una pista para proseguir con la investigación.

Fecha 3/9/2012

Fecha	**Sentido (E/S)**	**Cuerpo del mensaje**
2012-09-03T09:14:58	S	Hi,have you heard the news about **Stephen Fog**?
2012-09-03T09:17:02	E	Yeah,not surprised at all
2012-09-03T09:45:12	S	**Me too, he has always been a bad guy**
2012-09-03T11:10:09	E	Hi, check out **WhatsApp Messenger** for iPhone, Android, Nokia and BlackBerry. Get it now from http://whatsapp.com/dl/ and say good-bye to SMS

Tabla 37: Intercambios de mensajes del 03/09/2012

En este caso si bien no hay nada claramente delictivo, se menciona el nombre de una persona lo cual puede servir en la investigación, además el propietario del dispositivo investigado hace un comentario acerca de la misma y recibe una sugerencia de utilizar otro medio de comunicación.

Fecha 6/9/2012

Fecha	**Sentido (E/S)**	**Cuerpo del mensaje**
2012-09-06T09:00:19	E	Got it?
2012-09-06T09:00:39	S	What?
2012-09-06T09:02:03	E	Check it once again
2012-09-06T09:04:41	S	Got it on my position already
2012-09-06T09:06:20	E	Good! Let me know if there's any movement
2012-09-06T09:17:29	E	Black one?
2012-09-06T09:19:53	S	**Right,no movement yet. May be,15 minutes break?I have been watching for 6 hours already**
2012-09-06T09:20:55	E	Tell **Mike** to replace you
2012-09-06T09:22:41	S	He is ill today,I called him several hours ago.
2012-09-06T09:25:51	E	WTF?! Call **Andy** or **Jonny**
2012-09-06T09:27:58	S	Do you have Andy' s phone number? I lost my iPhone yesterday
2012-09-06T09:29:30	E	I'working with idiots! Forget it. I'll call him myself
2012-09-06T09:30:47	E	**Patrick** will be in 10 minutes
2012-09-06T09:34:44	S	Great!thanks for sorting it out
2012-09-06T09:35:07	E	As usual...
2012-09-06T09:37:27	S	Patrick is here.I'll be in an hour.in case of emergency just call me
2012-09-06T09:37:49	E	Ok

Tabla 38: Intercambios de mensajes del 06/09/2012

En esta conversación se observa una actividad a priori sospechosa, donde alguien vigila algo durante mucho tiempo, esta actividad podría catalogarse como un resultado positivo, además también aparecen otros nombres los cuales podrían ser utilizados como pistas para la investigación.

Observando el contenido de mensajes del día 11/2/2013 que no fue considerado sospechoso por no contar con una cantidad de comunicaciones fuera de lo habitual, se tiene que la conversación habla de una evasión fiscal.

Fecha 11/2/2013

Fecha	**Sentido (E/S)**	**Cuerpo del mensaje**
2013-02-11T09:59:27	S	Hi don't forget to print all the brochures
2013-02-11T10:01:20	E	Understood
2013-02-11T10:01:51	E	I'll do the best
2013-02-11T10:02:49	E	**I don't forget about tax evasion**

Tabla 39: Intercambios de mensajes del 11/2/2013

Lo anterior muestra un caso en que la herramienta pasa por alto algo que podría ser de mucha utilidad en el contexto de una investigación, por lo tanto sería un falso negativo.

Conclusiones

Con la ejecución de este caso se realizó una prueba completa de las funcionalidades más importantes que provee la herramienta, desde la carga de datos hasta su posterior análisis, generando los resultados que fueron interpretados y permitieron sacar algunas conclusiones.

Como punto a destacar, claramente se observa una reducción del volumen de información que el investigador debe considerar al iniciar el análisis, donde concretamente se observa que la herramienta señala como sospechosos aquellos intercambios de mensajes que se concentran en períodos de tiempo relativamente cortos. Donde el investigador deberá, dependiendo del contenido, discernir la utilidad de la información en el contexto de su investigación.

En general se considera que el prototipo desarrollado es muy satisfactorio, no solo porque cumplió con el principal objetivo para el que fue diseñado sino porque se le generaron extensiones que permiten entre otras cosas:

- cargar SMS, llamadas y contactos de una base sqlite.
- cargar SMS, eMails y contactos (de Facebook, WhatsApp y de la Agenda de Android) de archivos en formato CybOX.
- analizar los eventos utilizando operaciones parametrizables, dando más flexibilidad al investigador al momento de decidir como quiere considerar la evidencia a analizar.
- reproducir la secuencia de pasos realizada al utilizar la herramienta.

7. Trabajo Futuro

En este capítulo se describe el trabajo futuro que dejó el proyecto en el transcurso de sus etapas de análisis, diseño e implementación. Estas actividades se presentarán agrupadas por temas.

7.1. Análisis de comportamiento sospechoso

Como se mencionó en la sub-sección 3.5, el modelo estadístico *Series de tiempo*, parece apropiado para detectar comportamiento sospechoso, particularmente para determinar patrones de comportamiento. Por ejemplo, suponga el caso en que el propietario del smartphone se comunica habitualmente con un contacto entre las 9:00 y 18:00 horas, esto define un patrón de comunicación en lo que puede considerarse el horario típico de oficina, una comunicación a las 2:00 horas marca un cambio en ese patrón dando lugar a considerar esa comunicación como sospechosa.
Entonces, una línea a seguir como extensión de la detección de comportamiento sospechoso podría ser la implementación de un plugin que utilice como modelo estadístico las *Series de tiempo*.

7.2. Análisis relacional

Queda como trabajo futuro agregar algunas funcionalidades del tipo análisis relacional, esto es, mostrar vínculos entre el propietario y sus contactos y además poder vincular los contactos entre sí. Para esto último se necesita más información como ser grupos de SMSs, de WhatsApp, de Facebook, etc.
En la misma línea de análisis relacional se podría incorporar un grafo donde se mostraran los vínculos con el propietario y en las aristas del mismo se brindara información relacionada al tipo de comunicaciones que han tenido y la cantidad de las mismas (por ejemplo SMSs, llamadas, e-mails, etcétera).

Si bien se permiten cargar datos de diversas fuentes en el prototipo no se implementó ningún tipo de análisis relacional que permita vincular contactos de diversas fuentes. Esto sería de utilidad para poder dar indicios de qué personas (contactos de los dispositivos) podrían estar involucradas en la investigación llevada a cabo.

Permitir la consolidación de Contactos repetidos obtenidos de distintos orígenes, por ejemplo, Agenda del teléfono celular y contactos de Facebook.

7.3. Consideraciones forenses

Profundizar en el concepto de caso forense. Es decir, brindar la posibilidad de generar y administrar un *Caso*, donde cada caso puede tener asociado uno o varios dispositivos con sus respectivos timezone, donde además exista un investigador responsable y colaboradores. Toda esta información agregaría complejidad a la auditoría del sistema implementada. En otras palabras la herramienta manejaría la información necesaria para ayudar al investigador a presentar los resultados de la investigación y el proceso llevado a cabo con la misma ante la justicia o contraparte.

En esta línea lo único que se implementó fue la auditoría del sistema, esto es, la auditoría de los pasos realizados con el fin de poder reproducirlos en una etapa posterior y la comprobación de integridad de la base, plugins y core de la herramienta.

7.4. Soporte de otros tipos de datos

Como trabajo futuro queda pendiente brindar soporte y visualización para el contenido de los objetos del tipo *mensajería*, algunos de ellos son SMS, e-mails, mensajes de WhatsApp y chat de Facebook. La herramienta teniendo estos datos no solo podría indicar que eventos son considerados como sospechosos sino que podría brindar la posibilidad de mostrar el contenido de los mismos.

Otra característica interesante que quedó por fuera del desarrollo del prototipo fueron los datos geográficos que pueden obtenerse de los dispositivos móviles. Contar con ésta información permitiría correlacionar no solo tiempos sino también lugares, siendo aún más interesante si se investigan dos fuentes ya que analizando éstos datos se podría detectar que dos dispositivos se encontraban en el mismo lugar en un mismo momento.

7.5. Consideraciones de implementación

Se plantea como trabajo futuro, para mejorar la gestión de grandes volúmenes de datos, el cambiar la tecnología de base de datos utilizada por la aplicación, reemplazando el archivo *SQLite* [36] por un motor de base de datos, como por ejemplo lo son, *MySQL* [37] o *PostgreSQL* [38]. Este cambio, dado el diseño con el que fue concebida la herramienta desarrollada, impacta únicamente al módulo que se encarga de la persistencia del sistema.

Otro punto donde claramente hay campo fértil para avanzar es en las funcionalidades visuales que se le pueden agregar a la aplicación desarrollada, dado que en el contexto de una herramienta orientada al análisis como ésta, éstos son de mucha utilidad ya que permiten visualizar desde diferentes ángulos los datos o algún proceso que se haga sobre estos, mejorando la comprensión de los datos o para a partir de su interpretación inferir patrones, relaciones o vínculos que contribuyan en una investigación. Ejemplos concretos son, grafos para la representación de vínculos y líneas de tiempo con escalas de colores para distinguir intervalos de actividad.

7.6. Análisis de malware

Los enfoques de análisis que se pueden seguir dependiendo de como se considere al dispositivo involucrado dependen de si se estudia a éste como víctima de un ataque o como medio para cometer un ilícito, en este trabajo se usó el que considera al dispositivo como medio para cometer un ilícito. En esta sub-sección se pretende plantear líneas de trabajo que cubran el otro enfoque, el cual posiciona al dispositivo como víctima de un ataque.

En este nuevo enfoque se hace relevante el análisis de aplicaciones maliciosas. Si bien no se profundizó en este tema por no ser el foco de nuestro trabajo se hace un pequeño aporte al respecto porque, a priori, parece ser un interesante campo a explorar.
Durante la etapa del estado del arte se encontró literatura referida a la clasificación de aplicaciones Android, como benigna o maligna, por medio del análisis del archivo de manifiesto de éstas (AndroidManifest.xml) [39][40][41][42].

Concretamente, en base a lo definido en *"Detecting Android Malware by Analyzing Manifest Files"* [40] se pueden clasificar los componentes del archivo AndroidManifest.xml de una aplicación y asignarles una puntuación, si esa puntuación es mayor o igual a uno se considera que la aplicación es maligna.
En esta línea se propone realizar una adaptación al modelo propuesto por Dorothy E. Denning, para que

caracterice al archivo de manifiesto. Esa caracterización deberá considerar los componentes relevantes del archivo AndroidManifest.xml que serán puntuados, para luego usando el modelo estadístico *Operacional*, el que verifica si la puntuación excede el umbral fijado, poder determinar si se está o no frente a una aplicación maligna.

Lo anteriormente planteado aplica al análisis de una aplicación, conformando un primer caso de estudio. Se sabe que las aplicaciones en Android pueden colaborar entre sí [43][44], si bien no encontramos información al respecto, nos parece interesante pensar en la posibilidad de que aplicaciones "benignas" al ser combinadas en su uso den lugar a una aplicación "maligna". Lo que plantea un segundo caso de estudio, en el que, haciendo una abstracción, se considere que la aplicación tiene más de un archivo de manifiesto, tantos como aplicaciones colaboren en la determinación de la aplicación "maligna".

En conclusión se tiene que esta línea de trabajo puede ser una propuesta innovadora, ya que no solo se detectarían aplicaciones por si solas malignas, sino aquellas que son benignas pero combinadas las pueden determinar, y en consecuencia representar una amenaza para el dispositivo que las tenga instaladas.

8. Conclusiones

Este trabajo concierne al análisis forense de la evidencia digital obtenida de dispositivos móviles, etapa que se enmarca dentro del proceso de investigación forense del área *mobile forensics*. Dicha área, por ser relativamente nueva, no cuenta con el grado de estandarización de procedimientos forenses con el que se dispone en *computer forensics*. La explosión generalizada en el uso de dispositivos móviles y la presencia de éstos como objetos relevantes en procesos de investigación forense, generan la necesidad de contar con estándares específicos que los contemplen. Una respuesta inicial a esta necesidad la constituye la guía definida por el NIST [45].
En esta misma dirección, en "Smartphone Forensic Investigation Process Model" [46] se ilustra el desarrollo del modelo de proceso de análisis forense digital para Smartphones, se comparan metodologías forenses digitales y se propone un modelo de proceso sistemático de investigación forense para Smartphones. Ese modelo se adapta a la mayoría de las metodologías anteriores rectificando sus deficiencias y proponiendo algunos pasos extras que son necesarios para contemplar el avance de la tecnología.

Siendo el objetivo principal de este trabajo el desarrollo de una herramienta que asista al investigador forense en el análisis de la evidencia digital obtenida de dispositivos móviles, se implementó un prototipo que se diseña en base a un modelo de referencia reconocido, propuesto por Dorothy E. Denning para sistemas de detección de intrusiones. El modelo fue adaptado teniendo en cuenta que fue concebido para realizar detección de comportamiento sospechoso en tiempo real y ahora en el nuevo contexto se cuenta con el universo de los datos de estudio. Se considera que la elección de este modelo fue un gran acierto, ya que no solo le da al prototipo la capacidad de manejar evidencia encontrada en dispositivos móviles con Android, sino que lo posiciona como una herramienta capaz de manipular cualquier tipo de evidencia que registre la acción de un sujeto sobre un objeto en un momento dado, lo que es posible gracias a la abstracción con la que se representan los eventos.

Otro de los aportes del modelo de referencia incorporado al prototipo fue la capacidad de detectar comportamiento sospechoso. El que se obtiene al caracterizar los eventos considerando su cantidad y duración para determinar por medio de un modelo estadístico y en base a su historia previa si se está frente a un comportamiento sospechoso. Concretamente se proveen tres operaciones parametrizables que utilizan las métricas: contador de eventos y uso de recursos, y los modelos estadísticos: multivariado y media&desviación estándar.

Una de las características con la que se diseñó al prototipo fue que aceptara como insumo para analizar los propios resultados que genera. Esta característica junto a la de persistir los análisis que se ejecutan, con la que cuenta, da al investigador la posibilidad de seguir diferentes líneas de análisis, retomarlas y profundizarlas en cualquier momento. Se puede pensar como un árbol, donde cada nodo es el resultado de un análisis aplicado al nodo padre, siendo la raíz los datos que constituyen la potencial evidencia.
Potenciando lo anterior, el prototipo maneja el concepto de fuentes de información, lo que posibilita que se pueda distinguir la fuente de la que se obtuvo la potencial evidencia, permitiendo así realizar análisis que correlacionen eventos generados en diferentes dispositivos.

La escalabilidad para los métodos de análisis, carga y exportación de datos, es provista a través de una arquitectura basada en plugins, en la que se da una interfaz y una especificación de cómo deben diseñarse las extensiones para que puedan utilizarse desde el prototipo. Si bien puede suponerse, se hace explicito que para incorporar una nueva extensión no es necesario re-compilar el prototipo. El concretar este punto significó un hito importante no solo por las ventajas que da un diseño que permite adaptar el prototipo a cualquier tendencia en técnicas de análisis y representación de datos a consumir, sino porque ninguno

de los integrantes del grupo se encuentra orientado al área del desarrollo de software, lo que llevó a que muchas decisiones consumieron horas de investigación y discusión.

El prototipo es capaz de consumir información almacenada en el formato estándar CybOX, lo que se considera un punto importante a destacar dado que la mayoría de las aplicaciones que se consideraron relevantes para el trabajo usan su propio formato, que es propietario.
Además, puede consumir información almacenada en el formato SQLite de un modelo específico de dispositivo móvil. El por qué se desarrolló y ahora se destaca esta funcionalidad, se debe a la dificultad con la que nos encontramos al momento de determinar la existencia de los datos requeridos por el modelo propuesto, en el contexto de los dispositivos móviles con sistema operativo Android. No se encontró documentación oficial de Android donde se detallaran las estructuras de las bases de datos, vínculos entre tablas y mucho menos la semántica de los campos de las mismas. En fuentes no oficiales se encontraron algunas referencias que permitieron ver la heterogeneidad con la que los fabricantes (tanto de hardware como de versiones de Android) implementan las estructuras en las que persisten los datos, incluso habiendo diferencias para las distintas versiones de Android brindadas por un mismo fabricante. Dado que los dispositivos con Android almacenan los datos que se utilizan como potencial evidencia en archivos SQLite, encontramos que esta funcionalidad puede aportar en el sentido de ser una guía al momento de desarrollar un plugin que cargue datos desde un archivo SQLite, donde el desafío a enfrentar se centra en determinar la semántica usada por el fabricante del dispositivo. El desafío de determinar la semántica puede ser abordado, si no hay documentación al respecto, utilizando un conjunto de datos grande para inferir en que tablas están y como se relacionan esos datos, procedimiento que seguimos usando datos que recabamos de dispositivos propios y de conocidos, ya que disponer de un juego de datos relevantes es otro desafío a resolver.

En el campo forense uno de los aspectos importantes es poder reproducir, ante una contraparte o la justicia, a partir de potencial evidencia los pasos seguidos, por un investigador, que permitieron llegar a determinadas conclusiones. En este sentido el prototipo, dota al investigador con la posibilidad de ver que pasos siguió al utilizar la herramienta en su proceso de análisis de la potencial evidencia, por medio de un proceso automático en el que se registran las acciones realizadas con la herramienta.
Otro aspecto, no menos importante que el anterior, es garantizar de alguna manera que la potencial evidencia no fue alterada en el transcurso de la investigación, para esto la contribución que se hace, es controlar la integridad de la base de datos entre ejecuciones consecutivas de la aplicación para detectar, durante el proceso de investigación cuando la herramienta no es usada, si los datos que la herramienta maneja son alterados por fuera de ésta.

A nivel general, entre las dificultades y desafíos que se encontraron en el transcurso del trabajo, se destaca la carencia de literatura específica referida a la etapa de análisis dentro del proceso de análisis forense digital, la que generó mucho retrabajo debido a la ambigüedad del uso de la palabra “análisis” en este contexto. Lo anterior, junto con la inexperiencia (se estaba en la etapa de estudio del estado del arte), motivaron la lectura de muchos trabajos que terminaron siendo generales e incluso referidos a otra etapa del proceso (la examinación) ya que algunos autores la consideran parte de la etapa de análisis.

Para finalizar, se concluye que este trabajo deja como aportes un estudio del estado del arte del análisis de la evidencia de los dispositivos móviles, una propuesta de modelo para representar y caracterizar el comportamiento de sujetos sobre objetos, una herramienta extensible que asiste al investigador forense en el análisis de la evidencia; y varias líneas de trabajo futuro en las que profundizar el trabajo realizado.

Referencias

[1] Dorothy E. Denning: *An Intrusion-Detection Model*
`http://faculty.nps.edu/dedennin/publications/IDS%20model.pdf`
Último acceso: Noviembre 2015

[2] Juan Diana, José Varela: *Recolección de evidencia digital sobre dispositivos móviles : Android Inspector*
Proyecto de grado, Instituto de Computación, Facultad de Ingeniería, Universidad de la República. Agosto 2015

[3] International Data Corporation (IDC): *Smartphone OS Market Share, 2015 Q2*
`http://www.idc.com/prodserv/smartphone-os-market-share.jsp`
Último acceso: Noviembre 2015

[4] Kaspersky Lab: *¿Por qué los cibercriminales quieren TU smartphone? – Infografía*
`http://blog.kaspersky.es/hackers-smartphone-infografia/`
Último acceso: Noviembre 2015

[5] Virulist.com: *Evolución de las amenazas informáticas en el primer trimestre de 2014*
`http://www.viruslist.com/sp/analysis?pubid=207271252`
Último acceso: Noviembre 2015

[6] *National Institute of Standards and Technology*
`http://www.nist.gov/`
Último acceso: Noviembre 2015

[7] *The MITRE Corporation*
`http://www.mitre.org/`
Último acceso: Noviembre 2015

[8] *Digital Forensics Research Workshop*
`http://www.dfrws.org/`
Último acceso: Noviembre 2015

[9] Miguel López Delgado: *Análisis Forense Digital*
`www.oas.org/juridico/spanish/cyb_analisis_foren.pdf`
Último acceso: Noviembre 2015

[10] Eoghan Casey: *Digital Evidence and Computer Crime, third edition*
ISBN: 978-0-12-374268-1

[11] RFC 3227. D. Brezinski and T. Killalea. February 2002: *Guidelines for Evidence Collection and Archiving.*
`http://www.ietf.org/rfc/rfc3227.txt`
Último acceso: Noviembre 2015

[12] Aleksandar Valjarevic, H.S. Venter: *Analyses of the State-of-the-art Digital Forensic Investigation Process Models*
`http://www.satnac.org.za/proceedings/2012/papers/6.Management/11.pdf`
Último acceso: Noviembre 2015

[13] Germán González, Horacio Pérez: *Análisis de evidencia digital obtenida de dispositivos móviles - Estado del arte*
Proyecto de grado, Instituto de Computación, Facultad de Ingeniería, Universidad de la República. Diciembre 2015

[14] Research Workshop, August 2001, Utica, New York: *A Road Map for Digital Forensics*
`http://www.dfrws.org/2001/dfrws-rm-final.pdf`
Último acceso: Noviembre 2015

[15] International Organization for Standardization: *ISO/IEC 27042*
`https://www.iso.org/obp/ui/#iso:std:iso-iec:27042:ed-1:v1:en`
Último acceso: Noviembre 2015

[16] Simson Garfinkel, Alex Nelson and Joel Young. *A General Strategy for Differential Forensic Analysis*
`http://www.dfrws.org/2012/proceedings/DFRWS2012-6.pdf`
Último acceso: Noviembre 2015

[17] Christopher Hargreaves y Jonathan Patterson: *An automated timeline reconstruction approach for digital forensic investigations*
`http://www.dfrws.org/2012/proceedings/DFRWS2012-8.pdf`
Último acceso: Noviembre 2015

[18] Cellebrite: *UFED Physical Analyzer*
`http://www.cellebrite.com/es/Mobile-Forensics/Applications/ufed-physical-analyzer`
Último acceso: Noviembre 2015

[19] Autopsy: *Autopsy User Documentation*
`http://www.sleuthkit.org/autopsy/docs/user-docs/4.0/`
Ultimo acceso: Noviembre 2015

[20] The SleuthKit (TSK) & Autopsy: *Open Source Digital Forensics Tools*
`http://www.sleuthkit.org/`
Ultimo acceso: Noviembre 2015

[21] Autopsy: *Apache 2 License*
`http://www.sleuthkit.org/autopsy/apache2.php`
Ultimo acceso: Noviembre 2015

[22] MITRE: *Structured Threat Information eXpression (STIX)*
`http://stixproject.github.io/about/`
Último acceso: Noviembre 2015

[23] MITRE: *Cyber Observable eXpression (CybOX)*
`http://cyboxproject.github.io/about/`
Último acceso: Noviembre 2015

[24] *Oxygen Forensics*
`http://www.oxygen-forensic.com/`
Último acceso: Noviembre 2015

[25] MITRE: *The STIX Project*
`https://github.com/STIXProject/`
Último acceso: Noviembre 2015

[26] National Software Reference Library: *Project Web Site*
http://www.nsrl.nist.gov/
Último acceso: Noviembre 2015

[27] National Software Reference Library: *Digital Forensics XML (DFXML)*
http://www.nsrl.nist.gov/DFXML/fileobject.xsd
Último acceso: Noviembre 2015

[28] W. Alink, R.A.F. Bhoedjang, P.A. Boncz, A.P. de Vries: *XIRAF – XML-based indexing and querying for digital forensics*
http://www.dfrws.org/2006/proceedings/7-Alink.pdf
Último acceso: Noviembre 2015

[29] Sriram Raghavan: *Digital forensic research: current state of the art*
http://link.springer.com/content/pdf/10.1007%2Fs40012-012-0008-7.pdf
Último acceso: Noviembre 2015

[30] Andriller: *Smartphone Forensics Decoder*
https://andriller.com/decoders
Último acceso: Noviembre 2015

[31] Andrew Hoog: *Android Forensics: Investigation, Analysis and Mobile Security for Google Android*
ISBN-13: 978-1597496513 ISBN-10: 1597496510

[32] Friedrich-Alexander-University Erlangen-Nuremberg: *Safety-critical Consideration of Smartphone Usecases*
https://www1.informatik.uni-erlangen.de/filepool/thesis/android_usecases.pdf
Último acceso: Noviembre 2015

[33] International Journal of Computer Applications: *A Review of Anomaly based IntrsionDetection Systems*
http://www.ijcaonline.org/volume28/number7/pxc3874730.pdf
Último acceso: Noviembre 2015

[34] GitHub - PluginApplication: *A basic Java plugin architecture with dependency injection and singleton support.*
https://github.com/RovoMe/PluginApplication
Último acceso: Noviembre 2015

[35] Oxygen Forensics: *Demo Backups*
http://www.oxygen-forensic.com/en/download/devicebackups
Último acceso: Noviembre 2015

[36] *SQLite*
http://www.sqlite.org/
Último acceso: Noviembre 2015

[37] *MySQL*
http://www.mysql.com/
Último acceso: Noviembre 2015

[38] *PostgreSQL*
http://www.postgresql.org/
Último acceso: Noviembre 2015

[39] Franklin Tchakounté: *Permission-based Malware Detection Mechanisms on Android: Analysis and Perspectives*
http://www.scipublish.com/journals/CSSA/papers/download/3308-1050.pdf
Último acceso: Noviembre 2015

[40] Ryo Sato, Daiki Chiba y Shigeki Goto: *Detecting Android Malware by Analyzing Manifest Files*
http://journals.sfu.ca/apan/index.php/apan/article/download/110/pdf_59
Último acceso: Noviembre 2015

[41] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte, Pedrero, Javier Nieves, Pablo G.Bringas y Gonzalo Álvarez: *MAMA: Manifest Analysis for Malware Detection in Android*
http://paginaspersonales.deusto.es/isantos/publications/2013/sanz_2013_MAMA.pdf
Último acceso: Noviembre 2015

[42] Stephen Feldman, Dillon Stadther y Bing Wang: *Manilyzer: Automated Android Malware Detection through Manifest Analysis*
http://nlab.engr.uconn.edu/papers/manilyzer-camera-ready.pdf
Último acceso: Noviembre 2015

[43] Michael Grace, Yajin Zhou, Zhi Wang y Xuxian Jiang: *Systematic Detection of Capability Leaks in Stock Android Smartphones*
http://www.cs.fsu.edu/~zwang/files/NDSS12_Woodpecker.pdf
Último acceso: Noviembre 2015

[44] IBM: *Understanding security on Android*
http://www.ibm.com/developerworks/library/x-androidsecurity/
Último acceso: Noviembre 2015

[45] NIST: *Guidelines on Cell Phone Forensics*
http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-101r1.pdf
Último acceso: Noviembre 2015

[46] Archit Goel, Anurag Tyagi y Ankit Agarwal: *Smartphone Forensic Investigation Process Model* International Journal of Computer Science & Security (IJCSS) 6.5 (2012): 322-341.

9. Apéndice

9.1. Plugin de Análisis

Operaciones provistas por el plugin

El plugin desarrollado provee tres operaciones para detección de comportamiento sospechoso, las que se resumen en la tabla 40.

Identificador	Métrica que usa	Descripción
1	Cantidad de eventos	Esta operación analiza los registros de auditoría, considerando la cantidad de eventos[21] que tienen lugar en un intervalo de tiempo[22].
2	Duración de eventos	Esta operación analiza los registros de auditoría, considerando la duración de los eventos que tienen lugar en un intervalo de tiempo.
3	Cantidad y duración de eventos	Esta operación analiza los registros de auditoría, correlacionando las métricas cantidad y duración de eventos que tienen lugar en un intervalo de tiempo.

Tabla 40: Operaciones implementadas en el plugin de análisis

Cada operación cuenta con parámetros configurables por el investigador a través de un asistente, con el fin de aumentar las alternativas de análisis que éstas brindan de una manera relativamente sencilla.
En la figura 29 se presenta la interfaz del asistente de las operaciones mencionadas anteriormente, donde se pueden apreciar los parámetros que se permiten configurar.

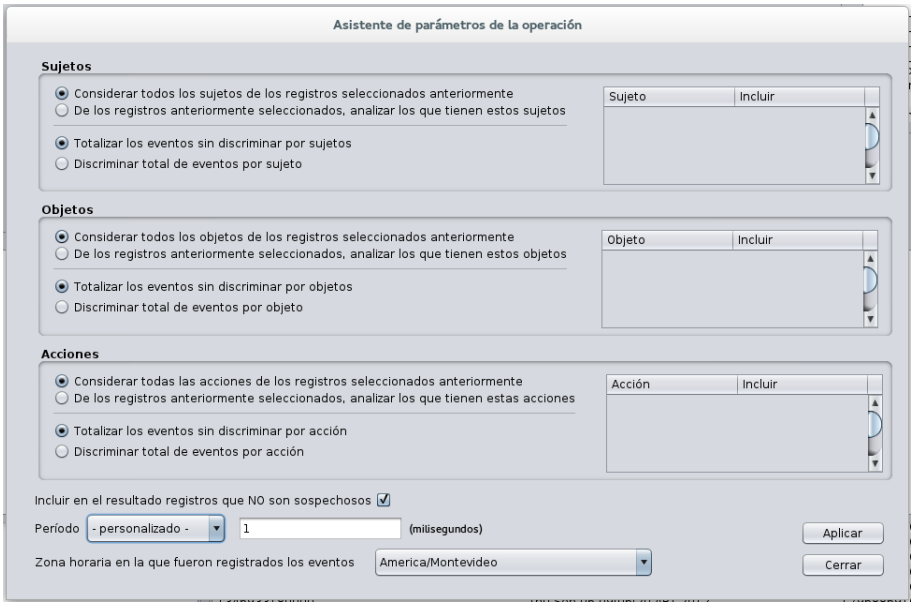

Figura 29: Asistente para configurar los parámetros de una operación del plugin de análisis

A continuación se describe qué son, para qué sirven y qué valores admiten éstos parámetros.

El Template como parámetro de una operación

La caracterización del comportamiento de sujetos sobre objetos se realiza utilizando la estructura denominada perfil, el que es creado a partir de un template.

[21] Por ejemplo: llamadas, e-mails y mensajes
[22] En milisegundos

El template establece la forma en la que los registros de auditoría se agruparan bajo un perfil.
El plugin desarrollado permite que se modifiquen algunos parámetros del template, permitiendo entonces que el investigador pueda definir variantes sobre los templates pre-definidos.
Dado que la definición del template puede resultar complicada, las primeras veces, el plugin brinda un asistente para cada operación, donde de una manera más sencilla se puede hacer la personalización del mismo.

A continuación se explica como definir el patron_sujeto, patron_objeto y patron_accion, que son los parámetros que presentan mayor dificultad al momento de comprender como utilizarlos.

El patron_sujeto tiene dos componentes, la primera denominada *matching-pattern* y la segunda *replacement-pattern*, teniendo como separador de las componentes el símbolo " <-", o sea que el patron_sujeto tiene la siguiente forma *matching-pattern <- replacement-pattern*.
matching-pattern: es el patrón con el que el sujeto del registro de auditoría debe tener una correspondencia para poder ser tenido en cuenta en el proceso de análisis. Para que un sujeto se corresponda con éste patrón tiene que ocurrir una de las siguientes opciones:

- el *matching-pattern* es un *****, lo que es interpretado como cualquier sujeto se corresponde con este patrón. En términos del asistente la opción que se corresponde es la que tiene el texto "Considerar todos los sujetos de los registros seleccionados anteriormente".
- el sujeto pertenece a una lista de sujetos dada de forma explicita, donde el separador entre elementos de la lista es el símbolo |. Por ejemplo, la lista con los sujetos ABC, DEF y GHI se denota como ABC|DEF|GHI. En términos del asistente la opción que se corresponde es la que tiene el texto "De los registros anteriormente seleccionados, analizar los que tienen estos sujetos" y se deben indicar los sujetos a considerar en la lista que aparece a su derecha.

replacement-pattern: patrón con el que se reemplazará el sujeto que tenga correspondencia con el *matching-pattern* definido. Desde el asistente se puede elegir entre dos opciones:

- Totalizar los eventos sin discriminar por sujetos: lo que indica que se quiere acumular bajo el mismo perfil cualquier registro de auditoría sin importar el sujeto. En este caso el valor del *replacement-pattern* es *****.
- Discriminar total de eventos por sujeto: lo que indica que se quiere acumular bajo el mismo perfil solo aquellos que corresponden a registros de auditoría que tienen igual sujeto. En este caso el valor del *replacement-pattern* es <**sujeto**>. La palabra reservada *<sujeto>* indica que el valor a utilizar es el del sujeto del registro de auditoría.

La siguiente figura muestra, para el sujeto, la selección realizada en el asistente correspondiente al *matching-pattern* **DE** y al *replacement-pattern* *****.

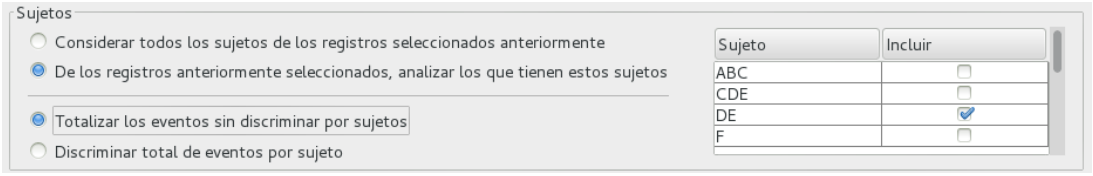

Figura 30: Sección del asistente donde se define el patron_sujeto de una operación del Plugin

Por último se presentan algunos ejemplos, donde en la primera columna se encuentra el patron_sujeto, en la siguiente el sujeto del registro de auditoría al que se le aplicará el patron_sujeto y en la última el resultado de dicha aplicación, si el registro es descartado en ésta columna se mostrará "- - - -".

patron_sujeto (template)	**sujeto** (registro de auditoría)	**sujeto** (perfil)
* <- <sujeto>	ABC	ABC
* <- *	ABC	*
ABC <- <sujeto>	ABC	ABC
ABC <- <sujeto>	DEF	- - - -
ABC <- *	DEF	- - - -
ABC|CDE|DE|F <- *	ACB	- - - -
ABC|CDE|DE|F <- *	ABC	*
ABC|CDE|DE|F <- *	DE	*
ABC|CDE|DE|F <- <sujeto>	DE	DE

Tabla 41: Resultado de transformar el sujeto de un registro de auditoría a través de un template

La definición tanto del patron_objeto como del patron_accion es análoga a la descripta para patron_sujeto, por lo que no se detallará. La palabra reservada en el *replacement-pattern* asociada a los objetos es *<objeto>* y a las acciones es *<accion>*.

Otros parámetros que se pueden configurar de una operación

Período: Corresponde al período medido en milisegundos para el cual se acumularán los registros dentro de un intervalo de un perfil, por ejemplo si el período es un día (86400000 milisegundos) se contabilizarán por intervalo los eventos de los registros de auditoría que correspondan al mismo día.

Unidad de tiempo: Es una manera alternativa, y más amigable, de definir el *Período*.
Por ejemplo podría tener definido las unidades de tiempo: día, semana, mes, año, etc. con sus respectivos valores en milisegundos, luego asignaría como unidad de tiempo la descripción de la unidad de tiempo a aplicar (día a modo de ejemplo) y el plugin obtendría automáticamente los milisegundos asociados a la misma.

Zona horaria en la que fueron registrados los eventos: Este dato es necesario para determinar correctamente los intervalos en los que se agruparan los valores de los registros de auditoría.

Incluir en el resultado registros que NO son sospechosos: Si bien el resultado del análisis es obtener aquellos registros que pueden evidenciar un comportamiento sospechoso, el plugin permite que se incluyan en los resultados aquellos registros que no lo evidencian.

d: Este parámetro está asociado al modelo que puede aplicarse a contador de eventos, intervalos de tiempo y acumulación de medición de recursos sobre un intervalo de tiempo fijo entre dos eventos relacionados. Según la inecuación de Chebyshev's, la probabilidad de que un valor caiga fuera del intervalo es a lo sumo $1/d^2$; ej. para d=4 es a lo sumo 0,0625.
Por defecto el valor es 4 y para dar una mejor idea de como influye en el cálculo del intervalo de confianza se muestra la ecuación en la cual interviene $media \pm d \times stdev$.