Universidad de las Ciencias Informáticas

Título: Sistema de registro del análisis de evidencias para un caso de informática forense

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autores:

Yanelis Avila Escuela

Alejandro Morales Boclis

Tutores:

MCs. Alina Surós Vicente

MCs. Yeneit Delgado Kios

La Habana, junio 2013

"El único hombre que no se equivoca es el que no hace nada"

Steve Jobs.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la UCI los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los días ____ del mes de _____ del año 2013.

Yanelis Avila Escuela	**Alejandro Morales Boclis**
Firma del Autor	Firma del Autor
MSc. Alina Surós Vicente	**MSc. Yeneit Delgado Kios**
Firma del Tutor	Firma del Tutor

Agradecimientos

Agradezco primeramente a Dios porque hoy puedo decir: "Eben-ezer", hasta aquí nos ha ayudado Dios. A mi compañero de tesis persona imprescindible en la realización de este trabajo, al igual que mí inigualable tutora, a Yeneit, qué decir de mi tutora Alina a quien le debo más que simples agradecimientos, gracias por la disposición de ayudarnos sin importar el día, la hora y el cansancio, gracias por confiar en nosotros.

Agradezco también desde lo más profundo de mi corazón a mis padres, por todo su esfuerzo y sacrificio, por sus suplicas y oraciones por mí, a quienes les debo todo lo que soy. A mi hermana Yanet, quien gracias a Dios la he tendido cerca de mí durante estos cinco años, a mi adorable sobrinita Elizabeth quien me devuelve la alegría tan solo con decir: "tía te quiero". A Julio C. Pompa, quien no solo ocupa un lugar muy importante en mi vida, sino que es parte de ella. A mi prima Odelis y a Arturo, gracias por todas sus bendiciones y a pesar de estar lejos siempre he podido contar con su apoyo, de igual modo a mi prima Onelvis, quien me habló siempre de esta carrera. A mi primo Adriel por toda su ayuda y apoyo.

A mis compañeros de grupo y a personas especiales que tuve la dicha de conocer, a Mailen gracias porque siempre pude y sé que puedo contar contigo, a Laritza por tener la paciencia de soportarme.

A todos mis hermanos en la fe, especialmente a Elizabeth, Aimet, Carlos, Aliana, Alayo y Lizandra, gracias a todos por ser de bendición en mi vida.

Yanelis

Agradecimientos

A mi madre Mercy en especial, por siempre apoyarme a lo largo de toda mi vida y ser el motor impulsor de toda mi carrera.

A Eliecer por ser más que un padre para mí y brindarme todo su apoyo.

A mi novia, mi compañera Adachely, por ser una persona especial y ayudarme en todo momento que lo necesité.

A mi hermano Elito, mi abuelo Abelardo, mi padre Alfredo y a toda mi familia por ser personas muy importantes para mí.

A mis amigos, en especial a Alain, Daniel, Gabriel, Liliett, Leinier por ser compañeros durante toda mi carrera.

A mis tutoras por brindarme toda su ayuda, apoyo y conocimientos.

A mi compañera de tesis Yanelis por batallar junto a mí para lograr la realización de este trabajo de diploma.

A todos mis profesores en especial a Joel Arencibia.

Alejandro

Dedicatoria

Dedico este trabajo a mis padres, Zoila Escuela Abreu y Jorge Luis Avila León, con el deseo de que siempre se sientan orgullosos de mí, así como yo de ellos.

Yanelis

Dedico este trabajo a Mercy Boclis Martínez por darme todo su apoyo.

Alejandro

Resumen

La presente investigación tiene como principal objetivo desarrollar un sistema de registro del análisis de evidencias para un caso de informática forense. Surge debido a la necesidad de mejorar el análisis de evidencias de un caso de informática forense en cuanto a su procesamiento y documentación. Para lograr el cumplimiento del objetivo de la investigación, se llevó a cabo un estudio de diferentes sistemas de análisis forense digital, enfocándose en las funcionalidades implementadas para la realización del mismo. Se investigaron las guías existentes a nivel mundial para el manejo de evidencias y un modelo formal para el análisis y la construcción de procedimientos forenses. Además se analizaron las herramientas y tecnologías a utilizar, así como la definición de los artefactos generados a partir de la metodología *Microsoft Solution Framework Agile* para el desarrollo de *software* ágil.

A partir de la realización de esta aplicación se obtuvo un conjunto de datos de delitos y componentes para distintos sistemas operativos que soportan la aplicación del modelo para la formalización del proceso de análisis forense digital permitiendo apoyar a los investigadores en el análisis y documentación de un caso de informática forense. Con el desarrollo del sistema de registro de evidencias se mejora el procesamiento de evidencias durante el análisis de un caso de informática forense. Este sistema podrá ser aplicado en entidades donde se realice análisis forense digital, como por ejemplo en la dirección de seguridad informática de la UCI o el laboratorio de criminalística de Cuba.

Índice general

Índice de tablas

Índice de figuras

Introducción

La aparición de la computadora y la posterior creación de la Internet han ampliado sustancialmente el acceso a la información, permitiendo la comunicación casi instantánea en todo el mundo. Con el creciente desarrollo computacional, el surgimiento y perfeccionamiento de las redes, se ha logrado la combinación de los conocimientos y habilidades en la obtención de herramientas y técnicas para alcanzar resultados exitosos en el cumplimiento de un objetivo. Un ejemplo de una rama de gran relevancia y que combina múltiples procedimientos para lograr objetivos que a veces parecen inalcanzables es la Ciencia Forense la cual proporciona los principios y técnicas que facilitan la investigación del delito criminal, es decir, cualquier principio o técnica que puede ser aplicada para identificar, recuperar, reconstruir o analizar la evidencia durante una investigación criminal. (García Martínez, 2001)

En esta última década la cantidad de delitos que involucran a las computadoras ha crecido, estimulando un incremento de compañías y productos que tienen como meta lograr el cumplimiento de la ley, basados en evidencia obtenida de las computadoras para determinar dónde, cuándo, cómo y por quién fueron ejecutados los delitos. Estos últimos pueden ir desde robo de la propiedad intelectual hasta lavado de dinero, fraude, destrucción de información confidencial, acoso sexual, amenazas vía correo electrónico, corrupción, pornografía, incluyendo pornografía infantil. Es aquí donde aparece un área nueva de la ciencia forense, llamada informática forense.

La informática forense surgió en la década de 1980 debido a los casos cada vez más comunes de *hardware* y *software* robados o falsificados, una consecuencia de la escalada de mercado de la computadora personal.

La informática o computación forense es la ciencia de adquirir, preservar, obtener y presentar datos que han sido procesados electrónicamente y guardados en un medio computacional (Noblett, 2000). Esta ciencia ha adquirido una gran importancia dentro del área de la información electrónica, esto es debido al aumento del valor de la información y su uso, al desarrollo de nuevos espacios donde es usada y a la extensa utilización de computadoras por parte de las compañías de negocios tradicionales.

Las nuevas leyes sobre delitos informáticos y la de firmas electrónicas y mensajes de datos abren procesalmente y definitivamente los medios probatorios informáticos. Las operaciones comerciales tienden claramente a reducir costos y ampliar mercados a través de las redes informáticas.

Es posible investigar (aun cuando Internet permite el anonimato y el uso de nombres falsos) quién es el dueño de sitios web, quiénes son los autores de determinados artículos y otros documentos enviados o publicados a través de redes. El rastreo depende en sí de quién y cómo realizó el ataque o cualquier otra acción.

Son igualmente investigables las modificaciones, alteraciones y otros manejos a las bases de datos de redes internas o externas, así como de cualquier sistema de redes y ataques internos. La destrucción de datos y la manipulación de los mismos también pueden rastrearse. Los hábitos de los usuarios en las computadoras y las actividades realizadas pueden ayudar a la reconstrucción de hechos, siendo posible saber todas las actividades realizadas en una determinada computadora.

Los archivos informáticos pueden guardar información sobre su autor, la compañía, fecha y otros datos de interés jurídico. Esta información es almacenada a espaldas del usuario, pudiendo determinarse en algunos casos en qué computadora fue redactado el archivo.

El objetivo de un análisis forense informático es realizar un proceso de búsqueda detallada y minuciosa para reconstruir a través de todos los medios el log de acontecimientos que tuvo lugar desde el mismo instante cuando el sistema estuvo en su estado íntegro hasta el momento de detección de un estado comprometedor. Esa labor debe ser llevada a cabo con máxima cautela y de forma detallada, asegurando que la evidencia se conserve intacta, de forma similar a los investigadores policiales que intentan mantener la escena del crimen inmune, hasta que se recogen todas las pruebas posibles.

Aunque existen varias herramientas para el análisis forense digital que aportan datos al análisis de un caso, el proceso es realizado de forma diferente dificultando que exista una manera normalizada de presentar un caso, por lo que surge la siguiente **situación problemática:** Los elementos asociados a la fase de análisis son los representativos en el uso de las herramientas diseñadas con este objetivo. El modo de realizar el análisis del dispositivo tiene diversas variantes, pudiendo aplicarse de forma diferente por cada investigador, siendo complejo para el equipo examinador el procesamiento de una manera estandarizada y no teniendo una guía de qué elementos de evidencia corresponden a cada uno de los pasos y qué detalles son importantes para analizar en cada caso. El artículo publicado en el año 2011, llamado: “Los retos de la informática forense en los próximos 10 años”, expresa que las herramientas son diseñadas para ayudar a los examinadores con elementos específicos de la evidencia, no para asistirlos en las investigaciones, implicando que el tiempo de análisis de un caso sea mayor. (Garfinkel, 2011)
Partiendo de la situación problemática existente, se plantea como **problema de la investigación**: ¿Cómo mejorar el análisis de evidencias de un caso de informática forense en cuanto a su procesamiento y documentación? El **objeto de estudio** se centra en: El proceso de análisis de evidencias durante la

investigación de un caso de informática forense. Se define como **objetivo general** de la investigación: Desarrollar un sistema de registro del análisis de evidencias para un caso de informática forense.

Para darle cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

- Elaborar el estado del arte del análisis de evidencias en la informática forense, las herramientas, tecnologías y metodologías a utilizar en el desarrollo de la solución.
- Realizar el análisis y diseño del sistema.
- Implementar la solución propuesta.
- Verificar que el sistema cumpla con las funcionalidades requeridas.

Tareas de investigación:

- Análisis del estado del arte de las soluciones para el análisis forense digital. (Yanelis y Alejandro)
- Análisis de guías y estándares para la recolección de evidencia digital. (Yanelis)
- Análisis de las herramientas, tecnologías, metodologías y lenguajes a utilizar en el desarrollo de la solución. (Yanelis, Alejandro)
- Identificación de los escenarios y requisitos de calidad de servicio. (Yanelis y Alejandro)
- Análisis y diseño de la solución. (Yanelis)
- Implementación de la solución propuesta. (Alejandro)
- Ejecución de pruebas a la solución para comprobar su correcto funcionamiento. (Yanelis)

Para un mejor desarrollo de la investigación se usaron los siguientes **métodos científicos**:

Método teórico:

Analítico-sintético: Facilita el entendimiento del fenómeno en el que se trabaja, es más útil la división de este en diferentes fases, y de esta forma descubrir sus características generales, lo que ayuda a seguir una correcta investigación.

Se utiliza este método con el objetivo de analizar y comprender la información y documentación de herramientas de análisis forense digital y extraer elementos importantes que se relacionen con el objeto de estudio.

Método empírico:

Entrevista: Es una conversación planificada entre el investigador y el entrevistado para obtener información.

Este método permitirá llevar a cabo entrevistas con varios profesores del centro para obtener sus experiencias en el tema, con el fin de aumentar el conocimiento en la rama informática forense.

Justificación de la investigación:

- Se han elaborado un conjunto de datos de delitos y componentes para distintos sistemas operativos que soportan la aplicación del modelo para la formalización del proceso de análisis forense digital basándose en la bibliografía actualizada en esta materia.
- El sistema resultante permitirá apoyar a los investigadores en el análisis y la documentación de un caso.
- La aplicación podrá ser utilizada en las entidades donde se realice análisis forense digital, como por ejemplo en la dirección de seguridad informática de la UCI o el laboratorio de criminalística de Cuba.

El presente documento ha sido estructurado en capítulos como se describe a continuación, para su mejor comprensión y estudio:

Capítulo 1. Fundamentación teórica: Contiene un análisis del estado del arte de las soluciones para análisis forense digital definido por estándares internacionales en esta materia, además de herramientas, lenguajes y metodologías a utilizar para el desarrollo de la solución.

Capítulo 2. Propuesta de solución: En este capítulo se realizará el análisis y diseño de la solución cumpliendo con la arquitectura y los patrones de diseño seleccionados para el desarrollo de la aplicación. Se generarán los artefactos necesarios durante el ciclo de vida del proceso de desarrollo, especificando las características del sistema desglosadas en escenarios y requisitos de calidad de servicios.

Capítulo 3. Implementación y prueba: Se incluye el diagrama de Componentes y de Despliegue, así como las pruebas unitarias y de caja negra para validar el cumplimiento de los requerimientos de la solución.

Capítulo 1. Fundamentación teórica

Introducción

En este capítulo se abordará el tema de informática forense destacando sus objetivos, usos, fases, así como el análisis de evidencia y las guías existentes para la manipulación de evidencias. Se realizará el estudio de los sistemas existentes para el análisis forense digital. Se analizarán las principales metodologías de desarrollo de *software*, tecnologías, lenguajes de programación y herramientas de desarrollo que se ajusten a la solución del problema.

1.1. Informática forense

En este epígrafe se abordarán los principales conceptos asociados al dominio del problema.

Informática forense: Según el FBI[1], la informática o computación forense es la ciencia de adquirir, preservar, obtener y presentar datos que han sido procesados electrónicamente y guardados en un medio computacional. (Noblett, 2000)

Computación forense: Disciplina de las ciencias forenses, que considerando las tareas propias asociadas con la evidencia, procura descubrir e interpretar la información en los medios informáticos para establecer los hechos y formular las hipótesis relacionadas con el caso; o disciplina científica y especializada que entendiendo los elementos propios de las tecnologías de los equipos de computación ofrece un análisis de la información residente en dichos equipos. (Cano Martínez, 2006)

Ciencia forense: Proporciona los principios y técnicas que facilitan la investigación del delito criminal, es decir, cualquier principio o técnica que puede ser aplicada para identificar, recuperar, reconstruir o analizar la evidencia durante una investigación criminal forma parte de la ciencia forense. (Acurio, 2007)

Evidencia digital: Según "*Guidelines for the Management of IT*[2] *Evidence*", la evidencia digital es *"*cualquier información, que sujeta a una intervención humana u otra semejante, ha sido extraída de un medio informático*"*. La evidencia computacional es única, cuando se la compara con otras formas de "evidencia documental". A diferencia de la documentación en papel, la evidencia computacional es frágil y una copia de un documento almacenado en un archivo es idéntica al original. Otro aspecto único de la evidencia computacional es el potencial de realizar copias no autorizadas de archivos, sin dejar rastro de que se realizó una copia. Esta situación crea problemas concernientes a la investigación del robo de

1 FBI: Oficina Federal de Investigación o Buró Federal de Investigación (en inglés: *Federal Bureau of Investigation*) es la principal rama de investigación del Departamento de Justicia de los Estados Unidos.

[2] IT: Tecnología de la investigación y las comunicaciones (en inglés: *Information Technology*).

secretos comerciales, como listas de clientes, material de investigación, fórmulas y *software* propietario. (Ghosh, 2004)

Delito informático: Son todas aquellas conductas ilícitas susceptibles de ser sancionadas por el derecho penal, que hacen uso indebido de cualquier medio informático. Implica actividades criminales que los países han tratado de encuadrar en figuras típicas de carácter tradicional, tales como robos, hurtos, fraudes, falsificaciones, perjuicios, estafas, sabotajes, pero siempre que involucre la informática de por medio para cometer la ilegalidad. (Tellez Valdéz, 2008)

Objetivos de la informática forense

La informática forense tiene tres objetivos, a saber:

- La compensación de los daños causados por los criminales o intrusos.
- La persecución y procesamiento judicial de los criminales.
- La creación y aplicación de medidas para prevenir casos similares.

Estos objetivos son logrados de varias formas, entre ellas, la principal es la recolección de evidencia. (León, 2001)

Usos de la informática forense

- **Prosecución criminal:** La evidencia incriminatoria puede ser usada para procesar una variedad de crímenes, incluyendo homicidios, fraude financiero, tráfico y venta de drogas, evasión de impuestos y pornografía.
- **Litigación civil:** Casos que tratan con fraude, discriminación, acoso, divorcio, pueden ser ayudados por la informática forense.
- **Investigación de seguros:** La evidencia encontrada en computadoras puede ayudar a las compañías de seguros a disminuir los costos de los reclamos por accidentes y compensaciones.
- **Temas corporativos:** Puede ser recolectada información en casos que tratan sobre acoso sexual, robo, mal uso o apropiación de información confidencial o propietaria, e incluso de espionaje industrial.
- **Mantenimiento de la ley:** La informática forense puede ser usada en la búsqueda inicial de órdenes judiciales, así como en la búsqueda de información una vez que se tiene la orden judicial para hacer la búsqueda exhaustiva. (León, 2001)

Fases del análisis forense digital

Durante el desarrollo de la investigación se analizó lo planteado por Eoghan Casey, NIJ[3], DFRWS[4] en el artículo:"*Digital Evidence: Representation and Assurance*" (Evidencia Digital: Representación y

[3] NIJ: Instituto Nacional de Justicia del departamento de los Estados Unidos (en inglés: *National Institute of Justice*).

[4] DFRWS: Taller de Investigación Forense Digital (en inglés: *Digital Forensics Research Workshop*).

aseguramiento) (Schatz, 2007) y lo expresado por Miguel L. Delgado en el libro: “Análisis forense digital” (López Delgado, 2007) como se muestra en la Figura 1.

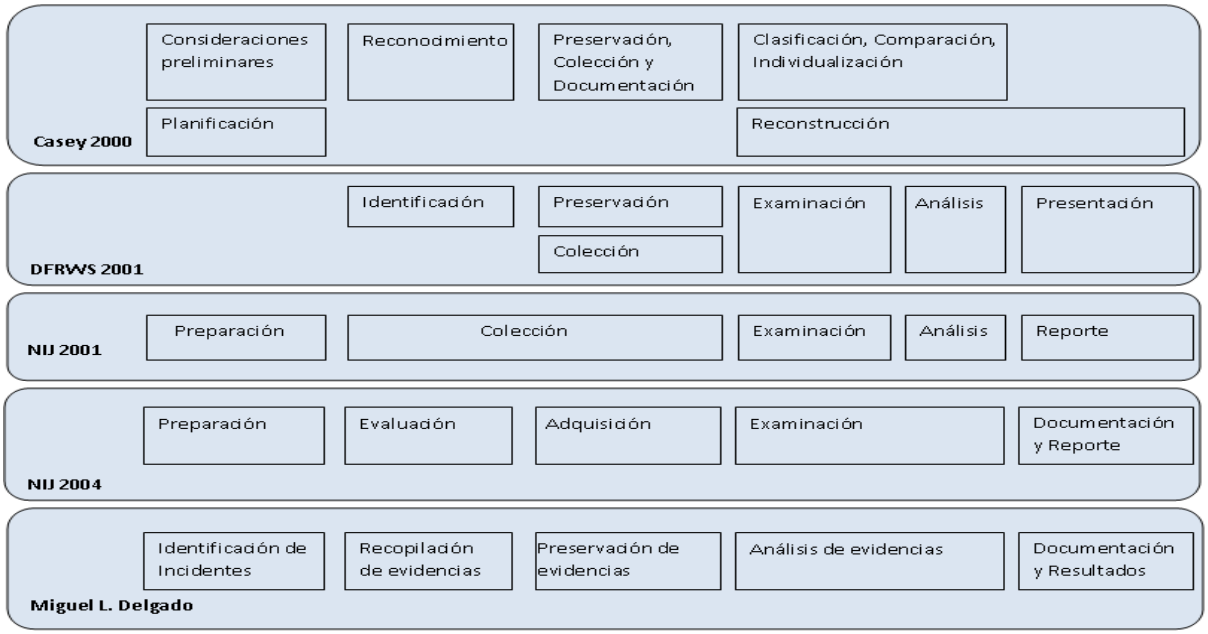


Figura 1. Fases del análisis forense digital

Todos los autores coinciden de manera general que la informática forense es el proceso de análisis forense digital amparado en un marco legal, por ello se tomó el proceso definido por Miguel Delgado porque se considera que abarca de forma general todas las propuestas.

Fases del análisis forense digital:

1ª. Identificación del incidente.

2ª. Recopilación de evidencias.

3ª. Preservación de la evidencia.

4ª. Análisis de la evidencia.

5ª. Documentación y presentación de los resultados.

Un **incidente de seguridad informática** puede considerarse como una violación o intento de violación de la política de seguridad, de la política de uso adecuado o de las buenas prácticas de utilización de los sistemas informáticos. (López Delgado, 2007)

1ª. Identificación del incidente

En esta primera fase se debe asegurar la integridad de la evidencia original, no se deben realizar modificaciones ni alteraciones sobre dicha evidencia. Adicionalmente, es preciso que el investigador o especialista se cuestione sobre la información obtenida en un sistema que se crea que está comprometido. Se deben establecer los procesos que se están ejecutando en el equipo ante un incidente

e identificar algún proceso extraño, o actividades poco usuales, pero para ello es preciso conocer la actividad normal del sistema. Por ejemplo, entre las principales actividades durante esta fase se deben consultar los registros del sistema en busca de avisos de fallos, accesos no autorizados, conexiones fallidas o cambios en archivos específicos del sistema.

2ª. Recopilación de evidencias

Se recomienda tomar apuntes detallados de todas las operaciones que se realice sobre los sistemas atacados, anotándose la fecha y hora de inicio y fin de cada uno de los pasos ejecutados, anotar también las características como números de serie de cada equipo, de sus componentes, del sistema operativo, etc. No escatimar en la recopilación de datos incluso hacer fotografías de los equipos y del entorno, porque cualquier evidencia puede ser definitiva. Es importante ser acompañado de otra persona durante el proceso de recopilación de evidencias.

3ª. Preservación de la evidencia

En este proceso, es imprescindible definir métodos adecuados para el almacenamiento y etiquetado de las evidencias. Como primer paso se debe realizar dos copias de las evidencias obtenidas, y generar una suma de comprobación de la integridad de cada copia mediante el empleo de funciones hash tales como MD5 o SHA1. Incluir estas firmas en la etiqueta de cada copia de la evidencia sobre el propio CD[5] o DVD[6], incluir además en el etiquetado la fecha y hora de creación de la copia, nombre de cada copia para distinguirlas claramente del original. Trasladar estos datos a otra etiqueta y pegarla en la caja contenedora del soporte, incluso será conveniente precintar el original para evitar su manipulación inadecuada. Otro aspecto a tener en cuenta es el proceso que se conoce como la cadena de custodia, donde se establecen las responsabilidades y controles de cada una de las personas que manipulan la evidencia. Se debe preparar un documento en el que se registren los datos personales de todos los implicados en el proceso de manipulación de las copias, desde que se tomaron hasta su almacenamiento.

4ª. Análisis de las evidencias

Una vez que se cuenta con todas las posibles evidencias de un incidente informático, se procede a su análisis. Esto involucra el análisis de log, la reconstrucción de información destruida, la recuperación de información oculta, etc. El análisis se debe llevar a cabo siguiendo una metodología rigurosa y científica, que permita la posterior reconstrucción de los pasos seguidos y que siempre conduzca al mismo resultado. También es importante preservar la integridad de la información que se está analizando, preferiblemente trabajando sobre una copia y no sobre el original para evitar alteraciones que podría invalidar la evidencia como una prueba aceptable ante la ley.

5ª. Documentación del incidente

---

[5] Disco Compacto (en inglés: *Compact Disc*), es un medio de almacenamiento de datos digitales como audio, imágenes, videos o texto plano.

[6] Disco de Video Digital (en inglés: *Digital Video Disc*) es un formato de almacenamiento digital de datos con mayor capacidad que el CD.

En la conclusión del análisis y durante su ejecución, se mantendrán informadas a las personas adecuadas de la organización, por lo que es interesante disponer de diversos métodos de comunicación. Tener preparados una serie de formularios y presentarlos tras la resolución del delito. (López Delgado, 2007)

Análisis de evidencias

La evidencia digital es una denominación usada de manera amplia para describir cualquier registro generado o almacenado en un sistema computacional que puede ser utilizado como prueba en un proceso legal. De acuerdo con el *Guidelines for the Management of IT Evidence* (Ghosh, 2004), la evidencia digital es cualquier información que, sujeta a una intervención humana u otra semejante, ha sido extraída de un medio informático. El documento establece que la evidencia digital puede dividirse en tres categorías:

- Registros almacenados en el equipo de tecnología informática (correos electrónicos, archivos de aplicaciones de ofimática, imágenes, etc.).
- Registros generados por los equipos de tecnología informática (registros de auditoría, de transacciones, de evento, etc.).
- Registros que parcialmente han sido generados y almacenados en los equipos de tecnología informática (hojas de cálculo financieras, consultas especializadas en bases de datos, vistas parciales de datos, etc.).

Cuando ha sucedido un incidente, generalmente, las personas involucradas en el crimen intentan manipular y alterar la evidencia digital, tratando de borrar cualquier rastro que pueda dar muestras del daño. Sin embargo, este problema es mitigado con algunas características que posee la evidencia digital. (Casey, 2004)

La evidencia digital puede ser duplicada de forma exacta y se puede sacar una copia para ser examinada como si fuera la original. Esto se hace comúnmente para no manejar los originales y evitar el riesgo de dañarlos. Actualmente, con las herramientas existentes, es muy fácil comparar la evidencia digital con su original y determinar si esta ha sido alterada.

La evidencia digital es muy difícil de eliminar, aun cuando un registro es borrado del disco duro de la computadora, y este ha sido inclusive formateado, es posible recuperarlo. Cuando los individuos involucrados en un crimen tratan de destruir la evidencia, existen copias que permanecen en otros sitios.

Criterios de admisibilidad

Existen cuatro criterios que se deben tener en cuenta para analizar la admisibilidad de la evidencia: la autenticidad, la confiabilidad, la completitud o suficiencia, y el apego y respeto por las leyes y reglas del poder judicial.

Autenticidad: Una evidencia digital será auténtica cuando se cumplan las siguientes condiciones:

- Se demuestra que dicha evidencia ha sido generada y registrada en el lugar de los hechos.

- La evidencia digital muestra 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. (Cano Martínez, 2003)

Confiabilidad: Los registros de eventos de seguridad son confiables si provienen de fuentes que son "creíbles y verificables" (Cano Martínez, 2003). Para probar esto, se debe contar con una arquitectura de computación en correcto funcionamiento, la cual demuestra que los log que genera tienen una forma confiable de ser identificados, recolectados, almacenados y verificados.

Una prueba digital es confiable si el "sistema que lo produjo no ha sido violado y estaba en correcto funcionamiento al momento de recibir, almacenar o generar la prueba". (Certain Jaramillo, y otros, 2005)

Suficiencia o completitud de las pruebas: Para que una prueba esté considerada dentro del criterio de la suficiencia debe estar completa. Para asegurar esto es necesario "contar con mecanismos que proporcionen integridad, sincronización y centralización" (Cano Martínez, 2003) para tener una visión completa de la situación. Para lograr lo anterior es necesario hacer una verdadera correlación de eventos, la cual puede ser manual o sistematizada.

Apego y respeto por las leyes y reglas del poder judicial: Este criterio se refiere a que la evidencia digital debe cumplir con los códigos de procedimientos y disposiciones legales del ordenamiento del país. (Cano Martínez, 2003)

Manipulación de la evidencia digital

A continuación se muestran los principios internacionales elaborados por la IOCE[7] los cuales constituyen los requisitos que se deben cumplir en cuanto a la manipulación de la evidencia digital:

- Cuando se maneje evidencia digital, deben ser aplicados todos los principios procedimentales y forenses generales.
- Al obtener evidencia digital, las acciones que se hayan tomado no pueden modificar esta evidencia.
- Cuando sea necesario que una persona acceda a la evidencia digital original, esa persona debe estar entrenada y calificada para este propósito.
- Todas las actividades relacionadas con obtención, conservación y transferencia de evidencia digital, deben estar completamente documentadas, preservadas y disponibles para revisión.
- El individuo es responsable por todas las acciones que realice con respecto al manejo de evidencia digital mientras esta esté bajo su cuidado.
- Cualquier agencia gubernamental que sea responsable de obtener, conservar y transferir evidencia digital, es responsable de cumplir con estos principios. (IOCE, 2013)

[7] IOCE: Organización Internacional de Evidencia Informática(en inglés: *International Organization on Computer Evidence*)

1.1.1. Gestión de la evidencia

Existen gran cantidad de guías y buenas prácticas que muestran cómo llevar a cabo la gestión de la evidencia digital presentando una serie de etapas para recuperar la mayor cantidad posible de fuentes digitales con el fin de asistir en la reconstrucción posterior de eventos.

Guías de mejores prácticas

A continuación se enuncian las guías existentes a nivel mundial de mejores prácticas en computación forense:

RFC 3227

El "RFC 3227: *Guidelines for Evidence Collection and Archiving*"[8] (Brezinski, 2002), escrito en febrero de 2002 por Dominique Brezinski y Tom Killalea, ingenieros del NWG[9], es un documento que provee una guía de alto nivel para recolectar y archivar datos relacionados con intrusiones. Muestra los principios durante la recolección de evidencia, las mejores prácticas para determinar la volatilidad de los datos, decidir qué recolectar, desarrollar la recolección y determinar cómo almacenar y documentar los datos. También explica algunos conceptos relacionados con las consideraciones de privacidad y legalidad.

Guía de la IOCE

La IOCE publicó "*Guidelines for the best practices in the forensic examination of digital technology*"[10] (IOCE, 2013). El documento provee una serie de estándares, principios de calidad y aproximaciones para la detección, prevención, recuperación, examinación y uso de la evidencia digital con fines forenses. Cubre los sistemas, procedimientos, personal, equipo y requerimientos de comodidad que se necesitan para todo el proceso forense de evidencia digital, desde examinar la escena del crimen hasta la presentación en la corte.

Investigación en la escena del crimen electrónico

El Departamento de Justicia de los Estados Unidos de América publicó "*Electronic Crime Scene Investigation: A Guide for First Responders*"[11] (Mukasey, 2008). Esta guía se enfoca en la identificación y recolección de evidencia. Se describen los tipos de dispositivos que se pueden encontrar y cuál puede ser la posible evidencia, las herramientas para investigar, el examen forense digital y clasificación de delitos.

Examen forense de evidencia digital

"Forensic Examination of Digital Evidence: A Guide for Law Enforcement"[12] (Justice, 2004) es una guía pensada para ser usada en el momento de examinar la evidencia digital. Describe el proceso de adquirir,

[8] Guía para recolectar y archivar evidencia.

[9] NWG: *Network Working Group.*

[10] Guía para las mejores prácticas en el examen forense de tecnología digital.

[11] Investigación en la Escena del Crimen Electrónico: Guía para Equipos de Respuesta.

[12] Examen Forense de Evidencia Digital: Una guía para la aplicación de la ley.

examinar y documentar la evidencia. Provee políticas y procedimientos con el fin de darle un buen trato a la evidencia.

Computación Forense - Parte 2: Mejores prácticas

La ISFS[13] creada en Hong Kong, publicó "*Computer Forensics-Part 2: Best Practices*"[14] (ISFS, 2004). Esta guía cubre los procedimientos y otros requerimientos necesarios involucrados en el proceso forense de evidencia digital, desde el examen de la escena del crimen hasta la presentación de los reportes en la corte. Recoge además la calidad en la computación forense y las consideraciones legales (orientado a la legislación de Hong Kong).

Guía de buenas prácticas para evidencia basada en computadores

La ACPO[15] del Reino Unido publicó "*Good Practice Guide For Computer Based Evidence*"[16] (ACPO, 1999). La policía creó este documento con el fin de ser usado por sus miembros como una guía de buenas prácticas para ocuparse de computadoras y de otros dispositivos electrónicos que puedan ser evidencia, detallándose en este documento los principios de la evidencia basada en computadoras.

Guía para el manejo de evidencia en IT (Guía Australia)

Standards Australia (Estándares de Australia) publicó "*Guidelines for the Management of IT Evidence*"[17] (Ghosh, 2004). Es una guía creada con el fin de asistir a las organizaciones para combatir el crimen electrónico. Establece puntos de referencia para la preservación y recolección de la evidencia digital. Detalla el ciclo de administración de evidencia como: diseño, producción, recolección, análisis, reporte y presentación y determinación de la relevancia de la evidencia.

Con el análisis de las guías de mejores prácticas en computación forense existentes a nivel mundial se logra obtener el orden de volatilidad de los datos y el desarrollo de la recolección, elementos fuera del alcance de este trabajo. También se tratan la RFC 3227 y ECSI en términos de decidir qué recolectar, cómo almacenar y documentar los datos, los cuales serán aplicados a la solución, aunque no complementan el proceso de análisis forense digital con los elementos a revisar en los dispositivos encontrados en la escena del crimen según el sistema operativo instalado.

1.1.2. Formalización de la evidencia digital

Los procedimientos de investigación forense se utilizan para detectar el alcance o la naturaleza del delito. En muchos casos, los procedimientos forenses utilizados están construidos de manera informal, lo que puede alterar la integridad de la investigación.

[13] ISFS: Sociedad de Seguridad Informática y Forense (en inglés: *Information Security and Forensic Society*)

[14] Computación Forense - Parte 2: Mejores Prácticas

[15] ACPO: Asociación de Jefes de Policía(en inglés: *Association of Chief Police Officers*)

[16] Guía de Buenas Prácticas para Evidencia basada en Computadoras

[17] Guía para el manejo de evidencia en IT

Ryan Leigland[18] propone un modelo formal para el análisis y la construcción de procedimientos forenses (Leigland, 2004), sobre dicho modelo estará fundamentada la investigación para realizar el registro de evidencia.

Descripción General

Procedimientos forenses: Se derivan de los delitos, con el fin de garantizar la capacidad de detectar un delito dado en un sistema operativo. Los procedimientos forenses se definen para ser independientes del sistema operativo.

Acciones forenses: Son específicas de un sistema operativo y se derivan de los procedimientos forenses. Un sistema computacional está dividido en pequeños componentes lógicos del sistema. En este contexto, **los componentes** son objetos abstractos, por ejemplo "ficheros de contraseñas", que no son específicos de una computadora o sistema operativo en particular. Una computadora o sistema operativo está compuesto por un conjunto de componentes.

Por tanto, dada la información acerca de un delito específico, el modelo permite la especificación de un conjunto de acciones como se muestra en la Figura 2.

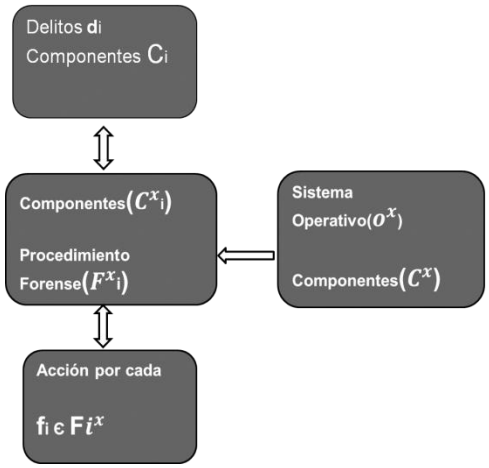


Figura 2. Creación del procedimiento para GNU\Linux. (Leigland, 2004)

Para una mejor comprensión del modelo, se describirá el siguiente ejemplo.

El primer paso en la creación del procedimiento de análisis forense digital es determinar la cobertura del delito, como se ejemplifica en la Tabla 1.

[18] Ryan Leigland recibió la categoría de máster en ciencias en ciencias de la computación en la Universidad de Idaho en el año 2004. Su principal interés es la investigación en seguridad informática y la informática forense.

Delito	Descripción
d1	Fraude económico
d2	Explotación infantil

Tabla 1. Delitos informáticos

Para correlacionar estos delitos con el sistema operativo es necesario crear la lista de componentes, como se ejemplifica en la Tabla 2.

Componente	Descripción
C1	Libreta de direcciones
C2	Correo electrónico
C3	Log de chat

Tabla 2. Componentes

Para cada uno de los delitos de la Tabla1 se genera la lista de componentes asociados, como se muestra en la Tabla 3.

Delito	Componentes
d1	C1, C2
d2	C2, C3

Tabla 3. Delitos-Componentes

Esta tabla muestra los componentes que pueden proporcionar pruebas de los distintos tipos de delitos. Si en el procedimiento generado se investiga los componentes adecuados, entonces se podrá investigar a fondo el delito en cuestión.

El siguiente paso es la creación de las primitivas, las cuales corresponden exactamente a un componente y representan la investigación de ese componente, como se muestra en la Tabla 4.

Primitiva	Descripción
f1	Examinar libreta de direcciones
f2	Examinar correo electrónico
f3	Examinar log de chat

Tabla 4. Primitivas forenses

Usando las primitivas de la Tabla 4 se genera el procedimiento forense necesario para detectar cada delito, como se muestra en la Tabla 5.

Procedimiento	Primitivas
FX1	f1, f2
FX2	f2, f3

Tabla 5. Procedimientos forenses

Usando el modelo y después del ejemplo, es posible crear y mantener procedimientos forenses con mayor precisión y exactitud. (Leigland, 2004)

Las guías de mejores prácticas permiten asistir al investigador en la recolección de evidencia digital, pero no particularizan los componentes que pueden constituir evidencias en un sistema operativo involucrado. Por tanto, el trabajo de investigación se basa en el modelo de formalización de la evidencia digital, el cual detalla los componentes a revisar según el tipo de delito y el sistema operativo y a la vez utiliza en conjunto las guías de mejores prácticas.

1.2. Sistemas para el análisis forense digital

En el artículo publicado en el año 2010 llamado "*Digital Forensic and Born-Digital Content in Cultural Heritage Collections*"[19], (Kirschenbaum, 2010) se realiza un análisis de las funcionalidades implementadas en las herramientas diseñadas para el procesamiento forense digital como se puede observar en la Figura 3.

La siguiente figura es un gráfico de barras donde en el eje de las ordenadas se muestran las principales herramientas de análisis forense digital y en el eje de las abscisas se representa la cantidad de funcionalidades de estas herramientas en escala de cinco. El color oscuro representado por la letra S, figura la cantidad de funcionalidades que sí están incluidas en la herramienta y el color claro representado por la letra P figura las funcionalidades parcialmente incluidas en la herramienta.

[19] Informática Forense y Contenidos de Origen Digital en las Colecciones del Patrimonio Cultural

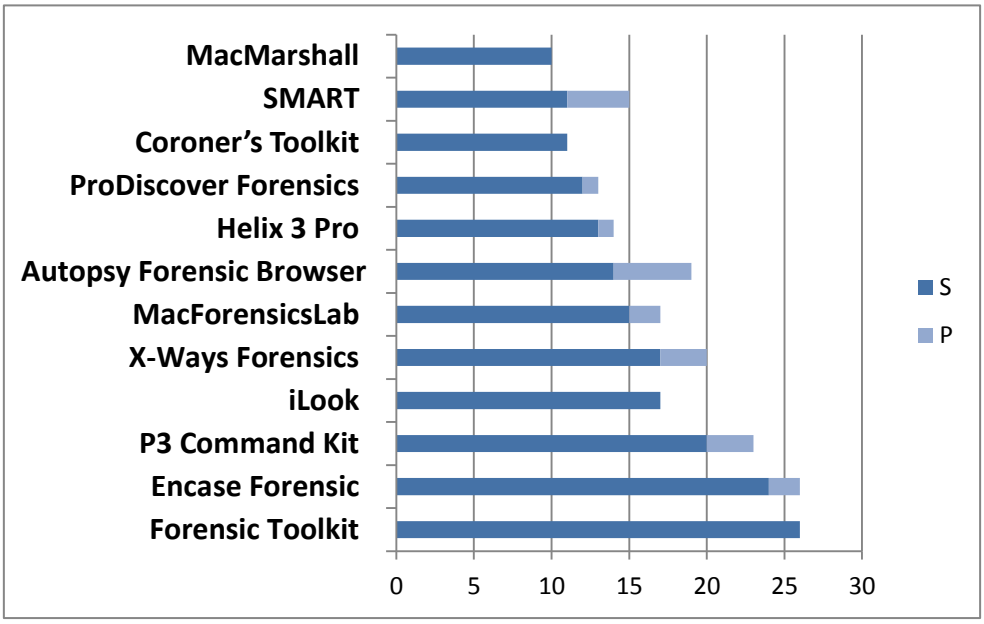


Figura 3. Análisis de sistemas similares

A continuación se analizarán las dos herramientas propietarias que más funcionalidades tienen: Forensic Toolkit y Encase Forensic, como herramientas libres Autopsy Forensic Browser y Helix, incluyéndose además, la herramienta DEFT.

1.2.1. Forensic Toolkit o FTK

La Caja de Herramientas Forenses (FTK) está basada en Windows y permite analizar el disco, sistema de archivos y datos de aplicación. Es compatible con sistemas de archivos FAT, NTFS y ext2/ 3. FTK crea un índice ordenado de las palabras en un sistema de archivos de modo que las búsquedas individuales son mucho más rápidas. Es compatible con muchos formatos de correo electrónico. (Accesdata, 2012)

FTK permite ver los archivos y directorios del sistema de archivos, recuperar archivos borrados, realizar búsquedas por palabras clave, ver todas las imágenes gráficas, las características del archivo de búsqueda en diferentes bases de datos y el uso de *hash* para identificar los archivos conocidos. En la versión FTK 4 incluye las siguientes características:

Manejo de casos: Se realiza de forma transparente en una base de datos Postgres.
Case overview[20]: Proporciona múltiples maneras de visualizar los datos presentes en las imágenes forenses añadidas al caso.
Análisis de datos volátiles y de memoria: Incluyendo procesos, *sockets*, archivos abiertos, parámetros, etc.

[20] Presentación de un caso

Opción de "Exportar información de archivos tipo *.LNK": Posibilita la generación de una plantilla con información valiosa sobre las actividades del usuario en el sistema operativo Windows, incluyendo la apertura de archivos en memorias USB y discos duros externos. Facilita la extracción de información relacionada con los accesos a carpetas compartidas.

El nuevo módulo de Cerberus para el análisis estático de binarios: Calcula una clasificación de riesgo basada en el uso de las comunicaciones de redes, persistencia, criptografía, ofuscación y funciones usadas por el binario, muy útil para casos de *malware*.

El nuevo módulo de visualización: Es extremadamente flexible, permitiendo la generación de gráficos basados en la información presente en el caso, como estadísticas de archivos seleccionados.

Procesamiento distribuido: Para casos grandes y para analizar más rápido, es posible habilitar el procesamiento distribuido hasta en cuatro máquinas esto puede significar un aumento importante de rendimiento. (Accesdata, 2012)

1.2.2. Encase

Es una herramienta desarrollada por *Guidance Software Inc.*, la cual permite asistir al especialista durante el análisis de un crimen digital. Dentro de sus principales características se encuentran:

Copiado comprimido de discos fuente: Encase emplea un estándar sin pérdida para crear copias comprimidas de los discos origen. Los archivos comprimidos resultantes pueden ser analizados, buscados y verificados de manera semejante a los originales, esta característica ahorra una cantidad importante de espacio en el disco de la computadora permitiendo trabajar en una gran diversidad de casos al mismo tiempo, examinando la evidencia y buscando en paralelo.

Búsqueda y análisis de múltiples partes de archivos adquiridos: Encase permite al examinador buscar y analizar múltiples partes de la evidencia, además de buscar todos los datos involucrados en un caso en un solo paso. La evidencia se clasifica, si está comprimida o no, y puede ser colocada en un disco duro y ser examinada en paralelo por el especialista. En varios casos la evidencia puede ser ensamblada en un disco duro o un servidor de red y también buscada mediante Encase en un solo paso.

Diferente capacidad de almacenamiento: Los datos pueden ser colocados en diferentes unidades, como discos duros IDE o SCSI, *drivers.* ZIP[21], y Jazz[22]. Los archivos pertenecientes a la evidencia pueden ser comprimidos o guardados en CD-ROM manteniendo su integridad forense intacta, estos archivos pueden ser utilizados directamente desde el CD-ROM evitando costos, recursos y tiempo de los especialistas. Encase permite al especialista ordenar los archivos de la evidencia de acuerdo a diferentes campos, incluyendo campos como marcas de tiempo (cuándo se creó, último acceso, última escritura), nombres de los archivos, firma de los archivos y extensiones.

[21] Es un sistema de disco removible de mediana capacidad, introducido en el mercado por la empresa Iomega en 1994.

[22] Es un sistema de almacenamiento de disco extraíble similar a *drivers* ZIP, presentado por la empresa Iomega en 1995.

Análisis compuesto del documento: Encase permite la recuperación de archivos internos y metadatos con la opción de montar directorios como un sistema virtual para la visualización de la estructura de estos directorios y sus archivos, incluyendo los datos del espacio no particionado.

Búsqueda automática y análisis de archivos de tipo zip y attachments de e-mail: La mayoría de las gráficas y de los archivos de texto comunes contienen una pequeña cantidad de *bytes* en el comienzo del sector los cuales constituyen una firma del archivo. Encase verifica esta firma para cada archivo contra una lista de firmas conocidas de extensiones de archivos. Si existe alguna discrepancia, como en el caso de que un sospechoso haya escondido un archivo o simplemente lo haya renombrado, Encase detecta automáticamente la identidad del archivo, e incluye en sus resultados un nuevo *ítem* con la bandera de firma descubierta, permitiendo al investigador darse cuenta de este detalle.

Análisis electrónico del rastro de intervención: Sellos de fecha, sellos de hora, registro de accesos y la actividad de comportamiento reciclado son a menudo puntos críticos de una investigación por computadora. Encase proporciona los medios prácticos de recuperar y de documentar esta información de una manera eficiente. Con la característica de ordenamiento, el análisis del contenido de archivos y la interfaz de Encase, virtualmente toda la información necesitada para un análisis de rastros se puede proporcionar en segundos.

Soporte de múltiples sistemas de archivo: Encase reconstruye los sistemas de archivos forenses en DOS, Microsoft Windows (todas las versiones), Macintosh (MFS, HFS, HFS+), GNU\Linux, UNIX (Sun, Open BSD), CD-ROM, y los sistemas de archivos DVD-R.

Con Encase un investigador va a ser capaz de ver, buscar y ordenar archivos desde estos discos concurridos con otros formatos en la misma investigación de una manera totalmente limpia y clara.

Vista de archivos y otros datos en el espacio no particionado: Encase provee una interfaz tipo Explorador de Windows y una vista del disco duro de origen, también permite ver los archivos borrados y todos los datos en el espacio no particionado.

Integración de reportes: Encase genera el reporte del proceso de la investigación forense como un estimado mostrando el caso incluido, la evidencia relevante, los comentarios del investigador, imágenes recuperadas, criterios de búsqueda y tiempo en el que se realizaron las búsquedas.

Visualizador integrado de imágenes con galería: Encase ofrece una vista completamente integrada que localiza automáticamente, extrae y despliega muchos archivos de imágenes como ‘gif’ y ‘jpg’ del disco. Si se selecciona la "vista de galería" se despliegan muchos formatos de imágenes conocidas, incluyendo imágenes eliminadas, en el caso de una vista pequeña.

El examinador puede escoger las imágenes relevantes para el caso e inmediatamente integrar todas las imágenes en el reporte de Encase. (Guidance, 2012)

1.2.3. DEFT

Deft Linux, acrónimo de *Digital Evidence & Forensic Toolkit*, es una distribución Linux italiana siendo Stefano Fratepietro el director y desarrollador de este proyecto basada en Ubuntu con *kernel* 2.6.35 y escritorio LXDE. Este sistema es distribuido como *Live* CD con una serie de aplicaciones forenses pensadas para la policía, investigadores, administradores de sistemas y especialistas forenses. Está dividido en dos, su entorno y herramientas *booteables* que recogen lo mejor del *software* libre para el análisis forense digital y DEFT Extra, un conjunto de herramientas gratuitas para análisis forense digital en entornos Windows. DEFT ofrece un excelente soporte para el *hardware* moderno y es fácil de usar. Como es una distribución basada en Ubuntu sigue la misma filosofía de un modelo de *software* libre y código abierto.

Deft Linux 7 está basado en Lubuntu 11.10 con *kernel* 3.0.0-12. Incluye características como: soporte para USB 3, imagen DVD instalable, analizador de iPhone, analizador de las copias de seguridad del iPhone, sumándose principalmente al análisis forense digital en teléfonos móviles añadiendo varias herramientas que permiten analizar los archivos y bases de datos utilizadas en teléfonos inteligentes de nueva generación. Se ha realizado una revisión completa de las funcionalidades de presentación de informes, incluyendo KeepNote, excelente herramienta que permite la organización lógica de las evidencias recogidas. En la nueva versión 7.2 se implementó un dispositivo virtual basado en VMWare 5 con soporte USB3, se actualizó el *kernel* a la versión 3.0.0-26, Autopsy 3 Beta 5, Log2tmeline se actualizó a la versión 0.65 al igual que Guymager a la versión 0.6.12-1, se añadió soporte para Vmfs, se implementaron algunas correcciones de espejo y está disponible el manual en inglés. (Deftlinux, 2013)

1.2.4. Helix

Posee la mayoría de las herramientas necesarias para realizar un análisis forense digital tanto de equipos como de imágenes de discos. Esta herramienta ofrece dos modos de funcionamiento, cuando es ejecutada permite elegir entre arrancar en un entorno Microsoft Windows o uno tipo GNU\Linux; en el primero de estos se dispone de un entorno con un conjunto de herramientas que permite principalmente interactuar con sistemas “vivos”, pudiendo recuperar la información volátil del sistema, mientras que en el arranque GNU\Linux se dispone de un sistema operativo completo con un núcleo modificado para conseguir una excelente detección de *hardware*, no realiza el montaje de particiones *swap*, ni ninguna otra operación sobre el disco duro del equipo sobre el que se arranque. Es ideal para el análisis de equipos “muertos”, sin que se modifiquen las evidencias pues montará los discos que encuentre en el sistema en modo solo lectura. (López Delgado, 2007)

1.2.5. Autopsy Forensic Browser

El navegador de Autopsy Forense es una interfaz gráfica para el *kit* Sleuth y otras herramientas de análisis. Se diseñó para ser una plataforma extensible que pueda ser una solución forense digital que incorpora módulos de conexión de los dos proyectos de código abierto y cerrado. Proporciona una aplicación basada en HTML que puede usarse para análisis forenses de los sistemas Windows y UNIX soportando sistemas de ficheros NTFS, FAT, UFS1/2, Ext2/3. (Sleuthkits, 2013). Su filosofía de funcionamiento se basa en:

- Un análisis de sistema "muerto", utilizando la herramienta en este caso desde otro sistema operativo y con el sistema a investigar en su soporte sin cargar. Los datos que se obtienen corresponden a la integridad de archivos, estructura de ficheros, log del sistema y datos borrados.
- Un análisis de sistema "vivo" -ocurre cuando se está analizando el sistema sospechoso mientras está funcionando-. Se analizan básicamente: procesos, memoria, fichero, etc. Después de confirmarse la amenaza, el sistema puede ser adquirido en una imagen con el objetivo de conservarlo sin corrupciones y así realizar análisis de sistema muerto.

Esta aplicación posee las siguientes técnicas de búsqueda de evidencias:

Listado del archivo: Analiza los archivos y los directorios, incluyendo los nombres de archivos suprimidos.

Contenido de archivos: Se puede ver en formato *raw*, *hex* o ASCII. Cuando se interpretan los datos, Autopsy los esteriliza para prevenir corrupción por parte del análisis del sistema local.

Bases de datos de hash: Los archivos son reconocidos como buenos o perjudiciales para el sistema basándose en la biblioteca de referencia del *software* y las bases de datos creadas por el usuario.

Clasificación de tipos de archivo por extensiones: Clasifica los archivos basándose en sus firmas internas para identificar extensiones conocidas. Autopsy puede también extraer solamente imágenes gráficas y comparar el tipo de archivo para identificar posibles cambios en la extensión para ocultarlos.

Línea de tiempo de la actividad del archivo: En algunos casos, tener una línea de tiempo de la actividad del archivo puede ayudar a identificar áreas de un sistema de ficheros que puedan contener evidencias. Autopsy puede crear la línea de tiempo que contienen los registros de: modificación, acceso, y cambios de fechas en los archivos.

Búsqueda de palabra clave: Las búsquedas de palabra clave de la imagen del sistema de ficheros se pueden realizar usando secuencias del ASCII y expresiones regulares. Las búsquedas se pueden realizar en la imagen completa del sistema de ficheros.

Análisis de los metadatos: Las estructuras de los metadatos contienen los detalles sobre archivos y directorios. Autopsy permite una visión de los detalles de cualquier estructura de los metadatos en el

sistema de ficheros. Esto es útil para recuperar el contenido eliminado. Autopsy buscará los directorios para identificar la trayectoria de los archivos.

Detalles de la imagen: Se pueden ver los detalles del sistema de ficheros, incluyendo la disposición en disco. Este modo proporciona información útil durante la recuperación de datos.

Gerencia del caso: Las investigaciones son organizadas por casos, que pueden contener uno o más anfitriones. Cada anfitrión se configura para tener su propia posición, ajuste de reloj y de zona horaria de modo que los tiempos examinados sean iguales a los del usuario original. Cada anfitrión puede contener una o más imágenes del sistema de ficheros a analizar.

Secuenciador de acontecimientos: Los acontecimientos se pueden agregar de los log de un IDS[23] o de un cortafuego. Autopsy clasifica los acontecimientos para poder determinar más fácilmente la secuencia de los acontecimientos del incidente.

Notas: Las notas se pueden clasificar en una base de datos organizados por anfitrión e investigador. Esto permite hacer notas rápidas sobre archivos y estructuras.

Integridad de la imagen: Es crucial asegurarse de que los archivos no están modificados. Autopsy, por defecto, generará un valor MD5 para todos los archivos. Se puede validar en cualquier momento la integridad de cualquier archivo que utiliza.

Informes: Autopsy permite crear los informes para los archivos y otras estructuras del sistema de ficheros.

Registros: Los registros de la intervención se crean en un caso, un anfitrión y un nivel del investigador para poder recordar fácilmente las acciones y los comandos ejecutados. (Sleuthkits, 2013)

Según el estudio de los sistemas para el análisis forense digital se puede concluir que existen gran cantidad de herramientas para recuperar evidencia, pudiéndose encontrar desde las más sencillas hasta las más sofisticadas que incluyen tanto *software* como dispositivos de *hardware*. En el caso de las herramientas analizadas como FTK y Encase la principal desventaja de estas consiste en que son *software* propietario y que su costo promedio es de $2.95 y $3.600 respectivamente (Kirschenbaum, 2010), mientras que las herramientas DEFT, *Helix* y *Autopsy Forensic Browser* son libres, pero siendo la recolección de evidencia una de las tareas más críticas, estas herramientas son diseñadas para ayudar a los examinadores con elementos específicos de la evidencia, no para asistirlos en las investigaciones, y no cuentan con una guía de qué elementos de evidencia corresponden a cada uno de los pasos y qué detalles son importantes para registrar en cada caso.

[23] Sistema de Detección de Intrusos (en inglés: *Intrusion Detection System,*).

1.3. Lenguajes, herramientas, tecnologías y metodologías

1.3.1. Lenguajes

Lenguaje de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevados por máquinas computadoras. Pueden usarse para crear programas que controlen el comportamiento físico o lógico de una máquina, para expresar algoritmos con precisión, o como modo interacción máquina-hombre. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. (Callao, 2008)

Lenguaje de programación Java

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de la compañía *Sun Microsystems.* Entre sus características fundamentales se encuentran:

Orientado a objetos: La filosofía de la Programación Orientada a Objetos, a diferencia de otros tipos de programación, es un paradigma que utiliza objetos como elementos fundamentales en la construcción de la solución.

Simple: Elimina la complejidad de los lenguajes como C y da paso al contexto de los lenguajes modernos orientados a objetos.

Robusto: Java se encarga internamente tanto de reservar la memoria como de liberarla, la liberación es completamente automática, ya que dispone del sistema de recogida de basura que se encarga de los objetos que ya no se utilizan. Además, proporciona la gestión de excepciones orientadas a objetos. En un programa de Java correctamente escrito, todos los errores de ejecución pueden y deben ser gestionados por el programa.

Seguro: Tiene políticas que evitan que se puedan codificar virus con este lenguaje. Existen muchas restricciones, que limitan lo que se puede o no hacer con los recursos críticos de una computadora.

Portable: Como el código compilado de Java (conocido como *bytecode*) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java.

Independiente de la arquitectura: El *bytecode* es interpretado por diferentes computadoras de igual manera, solo se necesita implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida.

Dinámico: No requiere que se compilen todas las clases de un programa para que este funcione. (Schildt, 2001)

Lenguaje de programación C#

C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en .NET *Framework*. Se puede

utilizar este lenguaje para crear aplicaciones cliente-servidor para Windows, servicios web XML, componentes distribuidos, aplicaciones de bases de datos, y muchas tareas más. (Network, 2013)

C# facilita el desarrollo de componentes de *software* a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen:

- Firmas de métodos encapsulados denominadas “delegados”, que permiten notificaciones de eventos con seguridad de tipos.
- Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Comentarios en línea de documentación XML.

El proceso de generación de C# es simple en comparación con el de C y C++, y es más flexible que en Java. No hay archivos de encabezado independientes, ni requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, estructuras, interfaces y eventos.

Ventajas

- Puede correr en diferentes plataformas, como en GNU\Linux y Microsoft Windows.
- Similitud de sentencias entre C++ y Java, hacen que sea sencillo el aprendizaje del mismo, lo que agiliza el tiempo de desarrollo de los proyectos.
- Existen innumerables librerías, propietarias y *open source* para el desarrollo con diferentes tipos de Bases de Datos. (MySQL, Oracle, PostgreSQL, etc.)
- Existen diferentes IDE[24] de programación (Visual Studio, MONO, C# Development) que ayudan directamente con la depuración de los programas desarrollados en el lenguaje C#. (Network, 2013)

Se selecciona el lenguaje de programación Java por ser multiplataforma, orientado a objetos, fácil de aprender y bien estructurado. Aunque Java fue adquirido por la compañía Oracle, el *software* se distribuye gratuitamente.

Lenguaje de modelado

Lenguaje Unificado de Modelado[25] es un lenguaje para especificar, visualizar construir y documentar los artefactos de los sistemas de *software*. Está destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. (Larman, 1999)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.

Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

[24] Entorno de Desarrollo Integrado
[25] UML por sus siglas en inglés

UML posibilita la captura, comunicación y nivelación de conocimiento estratégico, táctico y operacional para facilitar el incremento de valor, aumentando la calidad, reduciendo costos y reduciendo el tiempo de presentación; manejando riesgos y siendo proactivo para el posible aumento de complejidad o cambio. (Larman, 1999)

Se selecciona UML porque su utilización es importante debido a que permite especificar cuáles son las características del sistema antes de su construcción, ya que a partir de los modelos especificados se pueden construir los sistemas diseñados y los propios elementos gráficos sirven como documentación del sistema desarrollado. Por ser el lenguaje de modelado que se ha estudiado y aplicado durante la carrera.

1.3.2. Herramientas de desarrollo

Entorno de desarrollo

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de *Integrated Development Environment*) es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. (Alonso Velázquez, 2010)

NetBeans IDE es un entorno de desarrollo integrado, modular, de base estándar, escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (*framework*) para compilar cualquier tipo de aplicación. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. (Netbeans, 2013)

La plataforma NetBeans ofrece numerosos *frameworks* y muchas características que pueden ser de gran utilidad a la hora de desarrollar aplicaciones, entre las cuales se pueden resaltar:

- *Framework* para la creación de interfaces de usuario.
- Editor de datos.
- Interfaz de usuario para la personalización de la aplicación.
- Framework para la creación de asistentes.
- Sistema de datos que permite obtener información de diferentes orígenes de datos (FTP, CVS, XML o de una Base de Datos).
- Ayudas del sistema.
- Ayudas contextuales del sistema.

Además de estas características existen otras inherentes al uso de la plataforma, como son: un mejor aprovechamiento del tiempo de desarrollo, una mejor organización de la aplicación basada en estándares

y patrones estructurales y de diseño, una arquitectura consistente y robusta, y un mejor rendimiento en cuanto a tiempo de ejecución y optimización de recursos. (Netbeans, 2013)

Eclipse

Eclipse es una plataforma de desarrollo de código abierto basada en Java. Por sí misma, es simplemente un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada. Eclipse tiene un conjunto de complementos, incluidas las Herramientas de Desarrollo de Java. (Montero Garrido, 2001)

Características

- El principal objetivo de la plataforma Eclipse es proporcionar mecanismos y reglas que puedan ser seguidas por los fabricantes para integrar de manera transparente sus herramientas.
- Las diferentes herramientas que son instaladas en la plataforma Eclipse actúan sobre ficheros regulares que se almacenan en lo que se denomina espacios de trabajo, que es específico para el usuario.
- La plataforma Eclipse tiene interfaz gráfica, construida en base a un *workbench*[26] que proporciona toda la estructura y presenta una interfaz extensible al usuario.
- La plataforma tiene puntos de extensión y un API de proveedores que permite que nuevos tipos de repositorios puedan ser instalados.
- Proporciona mecanismos para definir y contribuir con documentación. (Montero Garrido, 2001)

Se selecciona NetBeans porque el lenguaje a utilizar para el desarrollo de la aplicación será Java con la JDK[27] y la JRE[28] y esta herramienta provee una serie de *frameworks* que agilizan el desarrollo.

Herramientas de modelado

La Ingeniería de *Software* Asistida por Computadora, conocidas sus herramientas como CASE, permite diseñar completamente el *software* de forma que provea ahorro de tiempo y recursos humanos en lugar de proceder a implementar el *software* directamente; utilizando esta ingeniería se evita, inclusive, realizar regresiones propias por concepto de diseño. (Larman, 1999)

Embarcadero ER/Studio

Embarcadero ER/Studio, herramienta para el modelado de datos, ayuda a las empresas a descubrir, documentar y reutilizar los datos. Con soporte completo a las bases de datos, los arquitectos de las mismas pueden realizar fácilmente ingeniería inversa.

Documenta y mejora las bases de datos existentes: ER/Studio provee una interfaz visual de fácil utilización para documentar, entender y publicar información acerca de las bases de datos existentes de

[26] Banco de trabajo

[27] Java *Development Kit*

[28] *Java Runtime Environment*

tal forma que puedan ser mejor controladas para soportar los objetivos del negocio. Con la opción de ingeniería inversa, permite al modelador de datos comparar y consolidar estructuras comunes de datos sin la necesidad de crear duplicaciones innecesarias.

Mejora la consistencia de los datos: Los trabajadores pueden invertir una cantidad significativa de horas buscando entre las fuentes de datos, investigando el significado de la información, y encontrando qué se ha estado utilizando adecuadamente. ER/Studio ayuda a los arquitectos de datos a definir y reutilizar los elementos comunes de los datos y elementos de modelado en los proyectos para establecer estándares en sus prácticas de modelado. Mediante el refuerzo de los estándares, y con la capacidad de analizar y documentar los elementos de datos, se pueden entender y utilizar mejor los datos, minimizar la redundancia y tener mayor consistencia.

Modela más que los datos: Con ER/Studio es posible utilizar el modelado para producir esquemas XML para asegurar los beneficios del modelado cuando se usa con aplicaciones y proyectos tales como arquitectura orientada a servicios. (Embarcadero, 2009)

Visual Paradigm

Se selecciona la herramienta profesional Visual Paradigm, porque soporta el ciclo de vida completo del desarrollo de *software*: análisis, diseño, construcción, pruebas y despliegue. El modelado del *software* ofrece una mejor y más rápida construcción del sistema, con mayor calidad y menor costo. Permite crear los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta provee el modelado de negocios, además de un generador de mapeo de objetos- relacionales para el lenguaje de programación Java y su integración con el NetBeans IDE anteriormente seleccionado.

Ofrece:

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Diseño centrado en funcionalidades y enfocado al negocio.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales Entornos de Desarrollo Integrado.
- Generación de código (modelo a código, diagrama a código).
- Generación de bases de datos (transformación de diagramas de Entidad-Relación en tablas de bases de datos).
- Disponibilidad en múltiples plataformas. (Paradigm, 2011)

Sistema gestor de bases de datos

Es un *software* específico, que permite crear y mantener una o varias bases de datos, asegurando mantener su integridad, confidencialidad y seguridad. Un sistema gestor de bases de datos está compuesto por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. (Estudioteca, 2013)

Los sistemas gestores de bases de datos deben cumplir algunos objetivos específicos como:

- Abstracción de la información.
- Independencia.
- Redundancia mínima.
- Consistencia.

MySQL

MySQL es un sistema de administración de bases de datos para bases de datos relacionales. Utiliza una arquitectura cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas y una gran variedad de interfaces de programación. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.

La condición de *open source* de MySQL, hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a internet.

El servidor de bases de datos MySQL es muy rápido, seguro y fácil de usar. Aunque se encuentra en desarrollo constante, este servidor ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad y seguridad hacen de él un servidor apropiado para acceder a bases de datos en internet.

En las últimas versiones se pueden destacar las siguientes características:

- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con tres archivos: de estructura, de datos y de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas. (Toledo Alma, y otros, 2013)

Ventajas

- Velocidad al realizar las operaciones, lo que lo hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

- Facilidad de configuración e instalación.
- Soporta gran variedad de sistemas operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad (Toledo Alma, y otros, 2013)

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD[29] y con su código fuente disponible libremente. (Martínez, 2010)

Su capacidad de comprimir y descomprimir sus datos sobre la marcha con un rápido sistema de compresión brinda la ventaja de ahorro de espacio de disco asignado y la lectura de datos se realiza con mayor rapidez. Cuenta con la capacidad de comprobar la integridad referencial, así como la de almacenar procedimientos en la propia base de datos, además de implementar el uso de subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz. Ofrece una estrategia de almacenamiento que permite trabajar con grandes volúmenes de datos gracias a su control de versionado concurrente.

Características

- Organiza los datos mediante un modelo objeto-relacional.
- Capaz de manejar procedimientos, rutinas complejas y reglas.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes.
- Cuenta con una API sumamente flexible propia para el trabajo con varios lenguajes de programación y procedurales como C, C++, Bash, Delphi, PL/Java, PL/Perl, PL/Tcl, PL/pgSQL, PL/Ruby, PL/PHP, PL/Python, PL/Scheme y PL/R.
- Ofrece transacciones que permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.
- Es altamente extensible, soporta operadores, funciones, métodos de acceso y tipos de datos declarados por el usuario; soporta además sobrecarga de operadores, sobrecarga de procedimientos, vistas materializables, particionamiento de tablas y datos.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de la información dentro de las bases de datos.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Las restricciones y disparadores tienen la función de mantener la integridad y consistencia en las bases de datos.

[29] Licencia BSD: es la licencia de *software* otorgada principalmente para los sistemas BSD(*Berkeley Software Distribution*)

- Usa una arquitectura cliente/servidor basada en un proceso por usuario; existe un proceso maestro que se ramifica para proporcionar conexiones adicionales por cada cliente que se intenta conectar a PostgreSQL. (Martínez, 2010)

Ventajas

- **Instalación**: No hay costo asociado a la licencia del *software*.
- **Soporte**: Existe una gran comunidad de profesionales y empresas que ofrecen soporte a PostgreSQL, de la cual el Grupo Global de Desarrollo de PostgreSQL es la principal.
- **Estabilidad y confiabilidad legendaria**: Miles de compañías reportan que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.
- **Extensible**: El código fuente está disponible para todos sin costo.
- **Multiplataforma**: Soporta alrededor de 34 plataformas incluyendo GNU\Linux y Unix en sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Microsoft Windows.
- **Posee herramientas gráficas de diseño y administración de bases de datos**: Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Data Architect).
- **Soporta funciones con privilegios por usuario**: Pueden definirse para ejecutarse con los derechos del usuario administrador o con los derechos de un usuario previamente definido, además retornan filas donde las salidas pueden tratarse como un conjunto de valores retornados por una consulta.
- **Soporta bloques de código**: Se ejecutan en el servidor y pueden ser escritos en diferentes lenguajes de programación con la potencia que presenta cada uno.
- **El máximo tamaño de bases de datos es ilimitado**: Depende del sistema de almacenamiento. (Martínez, 2010)

Se decide la utilización del gestor de bases de datos PostgreSQL por ser uno de los gestores más utilizados a nivel mundial, con uso libre de pago, para sistemas de gran escala o que necesitan un gran rendimiento en su desempeño. Posee muy buena integración con Java, el lenguaje de programación seleccionado. Su característica multiplataforma va acorde con la JRE para que sea desplegado en diversos ambientes de trabajo, así como, la explotación de su capacidad de comprimir y descomprimir datos con un rápido sistema de compresión. Ofrece la ventaja de ahorro de espacio de disco asignado y la lectura de datos se realiza con mayor rapidez brindando un desempeño mucho más fluido a nivel general en la aplicación.

JUnit

JUnit es un *framework* de código abierto para la automatización de las pruebas (tanto unitarias, como de integración) en los proyectos de *software*. El *framework* provee al usuario de herramientas, clases y

métodos que le facilitan la tarea de realizar pruebas en el sistema y así asegurar su consistencia y funcionalidad.

JUnit permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera, es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba, en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

En la actualidad, las herramientas de desarrollo como NetBeans y Eclipse cuentan con *plug-ins* que permiten que la generación de las plantillas necesarias para la creación de las pruebas de una clase Java se realice de manera automática, facilitando al programador enfocarse en la prueba y el resultado esperado, y dejando a la herramienta la creación de las clases que permiten coordinar las pruebas.

Ventajas:

- Las pruebas unitarias se automatizan.
- Si se realizan modificaciones en el código, se puede reutilizar este banco de pruebas para ver si nuevos requisitos interfieren en el funcionamiento antiguo.
- Los errores están más acotados y son más fáciles de localizar. (Usaola, 2006)

1.3.3. Tecnologías

Mapeo objeto-relacional

Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional y posibilitando el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

Ventajas

- **Abstracción de la base de datos:** Al utilizar un sistema ORM[30], lo que se consigue es separarse totalmente del sistema de base de datos que se utiliza, y así si en un futuro es necesario cambiar de motor de bases de datos, se tiene la seguridad de que este cambio no afectará al sistema, siendo el cambio más sencillo.
- **Reutilización:** Permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas.

[30] Mapeo objeto-relacional

- **Seguridad:** Los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser inyecciones SQL.
- **Mantenimiento del código:** Facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos.

Desventajas

- **Tiempo utilizado en el aprendizaje:** Este tipo de herramientas suelen ser complejas por lo que su correcta utilización requiere de tiempo para conocer su correcto funcionamiento.
- **Aplicaciones algo más lentas:** Esto es debido a que todas las consultas que se hagan sobre la base de datos, el sistema primero debe transformarlas al lenguaje propio de la herramienta, luego leer los registros y por último crear los objetos.

En la actualidad hay muchos tipos de framework que devuelven el mapeo objeto-relacional, según el lenguaje que se está trabajando. A continuación se muestra el framework a partir del lenguaje seleccionado para el desarrollo de la aplicación.

Hibernate

Es una herramienta de mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los *beans* de las entidades que permiten establecer estas relaciones.

Ventajas

- **Productividad:** Evita mucho el código tedioso de la capa de persistencia, permitiendo centrarse en la lógica de negocio. Permite una estrategia de desarrollo de aplicaciones *top-down* (empezar con el modelo de entidades) o *bottom-up* (trabajar con un modelo de datos existente).
- **Mantenibilidad:** Al tener pocas líneas de código permite que este sea más comprensible.
- **Rendimiento:** Actualizar las columnas que cambian en una sentencia *update* es más rápido en unas bases de datos, pero más lentas en otras. Toda esta lógica está embebida en el motor ORM. El motor está desarrollado por programadores con altos conocimientos de los SGBD[31] y la conectividad con Java.
- **Independencia de vendedor:** Una solución ORM abstrae del SGBD. Permite desarrollar en local con bases de datos ligeras sin implicación en el entorno de producción. (Ecured, 2012)

El uso de un ORM es una alternativa sumamente efectiva a la hora de trasladar el modelo conceptual (orientado a objetos) al esquema relacional nativo de las bases de datos SQL. Evita la inclusión de sentencias SQL embebidas en el código de la aplicación, lo que a su vez facilita la migración hacia otro sistema gestor de bases de datos. Incorpora una capa de abstracción entre el modelo relacional físico y la

[31] Sistema de Gestión de Bases de Datos

capa de negocios de la aplicación. Al ser realizada, en esta capa de manera automática la conversión de instrucciones orientadas a objetos a sentencias SQL, minimiza la ocurrencia de errores.

La selección de Hibernate está condicionada por el lenguaje de programación (Java) y el IDE (NetBeans) definidos para el desarrollo de la aplicación ya que esta es una herramienta de mapeo objeto-relacional de una potente integración con la plataforma Java.

1.3.4. Metodologías

Las metodologías de desarrollo de *software* surgieron por la necesidad de la industria del *software* de agilizarse y robustecerse al mismo tiempo. La evolución de estas ha estado indisolublemente ligada al crecimiento del *software* en sí mismo aportándole a este último diversas herramientas, las cuales permiten agilizar la documentación, otras el proceso de producción en sí mismo y otras todo el ciclo del desarrollo del *software*. A lo largo de los años han existido una gran variedad de estas, de las cuales se exponen para su valoración, solo las destacadas.

Programación Extrema

XP[32] es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de *software*, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas para enfrentar los cambios. XP se define como adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Extremeprograming, 2009)

Esta metodología tiene las características siguientes:

- Existe una estrecha comunicación entre los diseñadores y programadores, los integrantes del equipo y el cliente, y entre los mismos integrantes del equipo.
- El diseño del *software* es sencillo.
- El grupo de prueba permite obtener una mejor comunicación entre los usuarios y el cliente y así tomar sus experiencias.
- A medida que se proponen los cambios en el producto se van realizando lo más pronto posible y se reajustan el tiempo y costo en función de la evolución real del proyecto. (Extremeprograming, 2009)

Microsoft Solution Framework Agile

Microsoft Solution Framework Agile se caracteriza por ser de planificación adaptable a cambios y orientada a las personas. Su proceso introduce ideas importantes del *software* ágil, junto con los principios

[32] *Extreme Programming* por sus siglas en inglés

y prácticas de MSF para CMMI[33] como por ejemplo: admite una estrategia que utiliza múltiples iteraciones y un enfoque para la construcción de aplicaciones que se basa en escenarios. Esta metodología incorpora prácticas para el manejo de la calidad del servicio (el rendimiento y la seguridad) y facilita la automatización y la orientación que se necesita para apoyar el equipo de trabajo, incluyendo la gestión de configuración y de proyectos.
La definición, desarrollo y prueba del producto se realizan en pequeñas iteraciones provenientes del proceso incremental del proyecto, reduciéndose así el margen de error en las estimaciones y proporcionándose información rápida acerca de la exactitud de los planes del proyecto.

Esta metodología soporta 17 flujos de trabajo básicos, en los cuales se agrupan diferentes actividades, e incluye además cinco fases para el desarrollo y seguimiento del producto, estas son: Visión y Alcance, Planificación, Desarrollo, Estabilización e Implantación. (Microsoft, 2007)

MSF Ágil dispone de los tipos de elementos de trabajo siguientes:

- Escenario: Descripción de la necesidad o solicitud del usuario.
- Error: Defecto o desviación entre el comportamiento esperado y el comportamiento observado en el producto.
- Requisito de calidad de servicio: Material resultante esperado del producto final. El mismo puede ser un resultado, un problema resuelto o una característica, entre otros.
- Tarea: Acción independiente que debe realizar una persona o un grupo de personas.
- Riesgo: Evento o condición probable que puede dar resultados potencialmente negativos en el proyecto en el futuro. (Jiménez Ruiz, 2012)

Fases de la metodología MSF Ágil

Visión y Alcance: Trata de unir el proyecto/producto a las experiencias de otros trabajos, el equipo debe saber lo que el cliente desea y cómo lo desea. En esta fase se definen los líderes del proyecto, se hace la primera evaluación de riesgo del proyecto, se plantean cuáles son los objetivos que se persiguen con el trabajo y hasta dónde se quiere llegar con el proyecto, además se realiza la evaluación inicial de riesgos del proyecto.
Planificación: Es en esta fase cuando la mayor parte de la planeación para el proyecto es terminada. El equipo prepara las especificaciones funcionales, realiza el proceso de diseño de la solución y prepara los planes de trabajo, estimaciones de costos y cronogramas de los diferentes entregables del proyecto.
Desarrollo: Durante esta fase el equipo realiza la mayor parte de la construcción de los componentes (tanto documentación como código), sin embargo se puede realizar algún trabajo de desarrollo durante la etapa de estabilización en respuesta a los resultados de las pruebas. La infraestructura también es desarrollada durante esta fase.

[33] Modelo de Madurez de Capacidad Integrado

Estabilización: En esta fase se conducen pruebas sobre la solución, las pruebas de esta etapa enfatizan el uso y operación bajo condiciones realistas. El equipo se enfoca en priorizar y resolver errores y preparar la solución para el lanzamiento.

Implantación: En esta fase el equipo implanta la tecnología y los componentes utilizados por la solución, apoya el funcionamiento y la transición del proyecto, y obtiene la aprobación final del cliente.

Ventajas

- Crea una disciplina de análisis de riesgos que ayuda y evoluciona con el proyecto.
- Vinculación con el cliente como también orientado al trabajo en equipo.
- Tiene facilidad de soporte y mantenimiento.
- Es adaptable, se puede utilizar para proyectos de cualquier magnitud.
- Aplica mucho e incentiva el trabajo en equipo y la colaboración.
- Permite la reutilización de componentes ya desarrollados en ciclos anteriores.
- Es un modelo enfocado a los requerimientos del usuario.
- Es una metodología que se puede ajustar a equipos de trabajo compuestos por tres o más personas. (Jiménez Ruiz, 2012)

Estudiadas y expuestas las metodologías de desarrollo de *software*, se decide aplicar MSF Ágil por las siguientes características: es una metodología que provee mecanismos flexibles para aplicar soluciones adecuadas a los problemas tecnológicos y de negocios; MSF Ágil no es un marco estático y evoluciona respondiendo a los cambios en la tecnología y en los requerimientos de los proyectos; por la cantidad de personal cuatro en dicho proyecto, la premura del mismo y que los miembros del equipo pueden tomar diferentes roles durante el ciclo de vida del proyecto.

Conclusiones

El estudio de sistemas para análisis forense digital demostró que existen soluciones para el análisis de evidencia digital pero a la vez son muy costosas e implican que el tiempo de análisis de evidencia sea mayor por no estar diseñadas para asistir a investigadores con elementos específicos de la evidencia.

El análisis de las diferentes tecnologías, metodologías, lenguajes y herramientas según las necesidades de la solución determinó la utilización de la metodología MSF Ágil, como lenguaje de modelado UML 2.0, Visual Paradigm 8.0 como herramienta de modelado y PostgreSQL 9.1 como sistema gestor de bases de datos; como tecnología para el mapeo de objetos Hibernate 3.0 y el lenguaje de programación Java 1.6.0_25 usando el IDE de desarrollo NetBeans 7.2.1.

Capítulo 2. Propuesta de solución

Introducción

El presente capítulo está conformado según las fases de la metodología MSF Ágil para el desarrollo de *software*, declarándose la visión del proyecto mediante la descripción del producto a desarrollar, se ofrece además el levantamiento y descripción de los requerimientos funcionales en calidad de escenarios y los no funcionales o de calidad de servicios, la planeación de las iteraciones que se llevarán a cabo para desarrollar las funcionalidades y la arquitectura que guía el proceso de desarrollo de la aplicación.

2.1. Fase Visión

El sistema de registro del análisis de evidencias para un caso de informática forense será una herramienta que le permitirá al investigador forense tener una guía de qué elementos de evidencia corresponden a cada uno de los pasos y qué detalles son importantes para analizar en cada caso. Basado en el modelo propuesto por Ryan Leigland de la Universidad de Idaho en el año 2004, cuyo objetivo es permitir la formalización del procedimiento forense a un sistema informático comprometido, se identificaron los principales delitos y los componentes a analizar por cada delito asociado a un sistema operativo dado.

Para la aplicación del modelo propuesto se hace un análisis en diferentes fuentes bibliográficas que resumen y aúnan los componentes para los diversos sistemas operativos según el tipo de delito, como se muestra en la Tabla 6. En la cual se expresa el *Delito* que ha ocurrido, *SO* representa los sistemas operativos instalados en los dispositivos encontrados en la escena del crimen, *Componentes* los componentes a analizar por los investigadores y *Ubicación* el lugar exacto donde se encuentra para el sistema operativo correspondiente, señalando la *Bibliografía* que sustenta la afirmación. Es importante destacar que este es un resultado de la investigación realizada durante la elaboración del trabajo de diploma. La tabla completa se encuentra en el **Anexo I**.

Delito	SO	Componentes	Ubicación	Bibliografía
Investigación de muerte	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007) (García Martínez, 2001) (Solís, 2012)
		Calendario	/data/data/com.android.providers.calendar/	
		SMS&MMS	/data/data/com.android.providers.telephon/	
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	

		Caché	/data/data/com.google.android.location/	

Tabla 6. Delito-SO-Componente-Ubicación "Investigación de muerte"

La aplicación a desarrollar obtendrá estos elementos de la base de datos a la que es posible añadir nuevos delitos, componentes y sistemas operativos que no han sido previamente definidos. Permitirá registrar los datos importantes de las evidencias obtenidas a partir del análisis de un componente y brindará al investigador basado en esta información, el reporte a ser presentado en la corte y la línea de tiempo que permitirá ubicar los eventos de interés para correlacionar las evidencias encontradas.

2.2. Planificación

Luego de haber definido la Visión del sistema, se puede pasar a la fase de Planificación. En esta etapa del proyecto el equipo de trabajo analiza, identifica y además prioriza los requerimientos que describen la solución, conjuntamente con ello se generan algunos artefactos como son la lista de escenarios, así como la lista de requisitos de calidad de servicios, los cuales son utilizados para especificar los requisitos del *software* que sirven de guía para todo el proceso de desarrollo (Microsoft, 2007). Seguidamente se abordarán los artefactos generados en esta fase de acuerdo a la metodología MSF Ágil para el desarrollo de s*oftware*.

2.2.1. Escenarios del sistema

Los escenarios definen la interacción entre el usuario y el sistema. La descripción detallada de estos es de gran ayuda en la implementación de las funcionalidades del sistema, a continuación se muestra el listado de los escenarios identificados:

Lista de escenarios

- Autenticar usuario
- Gestionar caso
    - Crear nuevo caso
    - Abrir un caso
    - Guardar un caso
- Documentar los datos de la evidencia
- Mostrar reporte del caso
    - Mostrar línea de tiempo
    - Mostrar reporte del caso

- Gestionar investigador
    - Insertar investigador
    - Modificar investigador
    - Cambiar estado del investigador

2.2.2. Priorización de escenarios

La priorización de los escenarios es de gran importancia porque permite identificar cuáles son los más relevantes para darles un tratamiento diferenciado en el momento de implementarlos, los escenarios cuya prioridad sea alta con valor de 5 puntos, media 4 puntos y baja 3 puntos se implementarán en la primera, segunda y tercera iteración respectivamente, como se muestra en la Tabla 7.

Escenario	Prioridad
Gestionar caso	5
Documentar los datos de la evidencia	5
Mostrar reporte del caso	4
Gestionar investigador	3
Autenticar usuario	3

Tabla 7. Priorización de los escenarios

2.2.3. Plan de iteraciones

Una iteración es un conjunto de tareas programadas para ocurrir en un determinado período de tiempo. Determinar la duración de una iteración incluye tener en cuenta factores claves como: la fecha de entrega del proyecto, el tamaño de los escenarios y el tiempo de integración (Microsoft, 2007). MSF Ágil define las iteraciones como un período fijo de tiempo para programar tareas. Una iteración es generalmente un período de entre dos y seis semanas. Dichas iteraciones generalmente son numeradas consecutivamente y siguen una a otra de manera continua.

Dada la fecha de entrega orientada se ha tomado en consideración la realización de tres iteraciones, las cuales tomarían un tiempo de 16 semanas aproximadamente para su total desarrollo.

- **Iteración #1:** Se implementarán los escenarios de mayor prioridad en el sistema, siendo necesario un total de diez semanas aproximadamente.
- **Iteración #2:** Se implementarán los escenarios de prioridad media en el sistema, siendo necesario un total de cuatro semanas aproximadamente.
- **Iteración #3:** Se implementarán los escenarios de prioridad baja en el sistema, siendo necesario un total de dos semanas aproximadamente.

La Tabla 8 muestra la planificación de los escenarios identificados en el sistema.

No	Escenarios	Prioridad	Riesgo	Esfuerzo (días)	Iteración
1	Gestionar caso	5	Alta	35	1
2	Documentar los datos de la evidencia	5	Alta	35	1
3	Mostrar reporte del caso	4	Media	28	2
4	Gestionar investigador	3	Baja	14	3
5	Autenticar usuario	3	Baja	7	3

Tabla 8. Planificación de los escenarios

2.2.4. Requisitos de calidad de servicios

Los requisitos de calidad de servicios, conocidos así en la metodología MSF Ágil, son las especificaciones que el sistema debe cumplir. Se clasifican en diferentes tipos como: soporte, usabilidad, rendimiento, fiabilidad, eficiencia, seguridad (Microsoft, 2007). A continuación se muestra la lista de los requisitos de calidad de servicios que se identificaron para el desarrollo del sistema.

Usabilidad

- El sistema podrá ser utilizado por usuarios que tengan un conocimiento básico en sistemas computacionales.
- La interfaz debe ser amigable y lo menos compleja posible con el fin de facilitar el manejo de las funcionalidades.

Seguridad

- El sistema deberá denegar el acceso a usuarios no autorizados.
- Solo se accederá a la base de datos desde la aplicación, nunca directamente desde el gestor de bases de datos.
- Cuando el usuario esté autenticado como administrador tendrá acceso a todas las funcionalidades implementadas.
- Cuando el usuario esté autenticado como investigador no tendrá acceso a la funcionalidad Gestionar investigador.

Restricciones de diseño

- El sistema debe implementarse usando el lenguaje Java, utilizando el IDE NetBeans.
- El sistema gestor de bases de datos a emplearse en la aplicación debe ser PostgreSQL.

2.2.5. Descripción de los escenarios

A continuación se muestra la Tabla 9 con la descripción del escenario **Crear un nuevo caso**, los restantes escenarios se encuentran documentados en el **Anexo II**.

Nombre del Escenario: Crear un nuevo caso		***Identificador:*** ES1
Objetivo del Escenario: Permitir crear un nuevo caso		
Persona: Investigador		
Iteración: 1ra	***Prioridad:*** 5	***Complejidad:*** Alta
Descripción: El investigador accede a la aplicación y selecciona la opción Nuevo Caso, el sistema muestra un listado de delitos previamente almacenados en la base de datos y un listado de sistemas operativos. El investigador selecciona el delito a investigar y el o los sistemas operativos involucrados, en caso de no ser alguno de los mostrados debe seleccionar la opción Otros. Luego selecciona la opción Iniciar y debe guardar el nuevo caso en el sistema.		

Tabla 9. Descripción del escenario Crear un nuevo caso

2.3. Arquitectura

La necesidad del manejo de la arquitectura de un sistema de *software* nace con los sistemas de mediana o gran envergadura, que se proponen como solución para un problema determinado. En la medida que los sistemas de *software* crecen en complejidad, bien sea por número de requerimientos o por el impacto de los mismos, se hace necesario establecer medios para el manejo de esta complejidad. En general, la técnica es descomponer el sistema en piezas que agrupan aspectos específicos del mismo producto de un proceso de abstracción y que al organizarse de cierta manera constituyen la base de la solución de un problema en particular. (Camacho, 2004)

Arquitectura por capas

El patrón de arquitectura por capas es una de las técnicas más comunes que los arquitectos de *software* utilizan para dividir sistemas de *software*. Dividir un sistema en capas tiene importantes beneficios:

- Se puede entender una capa como un todo sin considerar las otras.
- Las capas se pueden sustituir con implementaciones alternativas de los mismos servicios básicos.
- Se minimizan dependencias entre capas.
- Las capas posibilitan la estandarización de servicios.
- Luego de tener una capa construida, puede ser utilizada por muchos servicios de mayor nivel.

Para el desarrollo de la solución se implementó un patrón n-capas, definiendo tres capas: Presentación, Negocio y Acceso_Datos, como se muestra en la Figura 4 y se detalla a continuación.

Capa de Presentación: Contiene todas las interfaces que presenta el sistema al usuario con las cuales va a interactuar y sus correspondientes validaciones.

Capa de Negocio: Esta capa engloba las clases necesarias para realizar de manera correcta todas las funcionalidades específicas del negocio del sistema que dan solución a sus requerimientos.

Capa de Acceso_Datos: Contiene las clases generadas a través del ORM Hibernate y las que permiten gestionar las operaciones sobre la base de datos, así como, los objetos de la base de datos.

La Figura 4 muestra la arquitectura del sistema.

Figura 4. Arquitectura del sistema

2.3.1. Patrones de diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de *software*, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular.

Son menores en escala que los patrones arquitectónicos, y tienden a ser independientes de los lenguajes y paradigmas de programación. Su aplicación no tiene efectos en la estructura fundamental del sistema, pero sí sobre la de un subsistema, debido a que especifica a un mayor nivel de detalle, sin llegar a la implementación, el comportamiento de los componentes del subsistema. (Camacho, 2004)

Patrones GRASP

Los patrones GRASP[34] describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Gamma, 2003). Se dividen en varias categorías, y las que se usan para desarrollar la aplicación son las siguientes:

Patrón Experto: Asignar responsabilidades de la forma más eficiente.

Se le asignarán las principales responsabilidades al experto en la información o la clase que cuenta con la información necesaria para cumplir dichas responsabilidades. A continuación se muestra en la Figura 5 el uso de este patrón donde se evidencia que la clase Investigador es la que contiene la información necesaria referida a cada investigador.

[34] Patrones generales de *software* para asignación de responsabilidades (en inglés: *General Responsibility Assignment Software Patterns)*

```
public class Investigador  implements java.io.Serializable {


     private int idInvestigador;
     private Boolean esAdministrador;
     private Boolean esInvestigador;
     private Date fechaCreacion;
     private Boolean estado;
     private String nombre;
     private String primerApellido;
     private String segundoApellido;
     private String ci;
     private String usuario;
     private String contrasena;
     private Set<DatosEvidencia> datosEvidencias = new HashSet<DatosEvidencia>(0);
     private Set<Caso> casos = new HashSet<Caso>(0);

    public Investigador()
    {
    }
```

Figura 5. Patrón Experto

Patrón Creador: Responsable de crear una nueva instancia de alguna clase.

El Patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Se hace uso de este patrón cuando es necesario que las clases creen instancias con el nivel necesario de información para acceder a los datos almacenados, de acuerdo a la ejecución de una determinada acción. Las clases N_Caso y A_Caso son ejemplos donde se utiliza este patrón ya que estas tienen la responsabilidad de instanciar un caso, como se ejemplifica en la Figura 6.

```
public class N_Caso {

    public N_Caso() {
    }

     public void Crear_Caso( Date fechaCreacion,String nombre, boolean estado,
                             Investigador id_inv, Delito id_delit)
    {
        A_Caso acceso=new A_Caso();
        Caso caso= new Caso(id_delit, id_inv, fechaCreacion, nombre, estado);
        acceso.Insertar_Caso(caso);

    }
```

Figura 6. Patrón Creador

Patrón Bajo Acoplamiento: Soportar bajas dependencias.

El Patrón Bajo Acoplamiento es un principio que se debe recordar durante las decisiones de diseño, porque es el encargado de disminuir las dependencias de una clase con las demás. Soporta el diseño de clases más independientes y esto reduce el impacto de los cambios que puedan producirse en la inserción de nuevas características o el mantenimiento del sistema a nivel de desarrollo. En el diseño propuesto el acceso a datos sólo depende de la clase controladora que la usa, así se pueden realizar cambios en cada clase de forma independiente.

Patrón Alta Cohesión: Mantiene la complejidad manejable.

El Patrón Alta Cohesión evita asignar demasiadas “responsabilidades” a las clases. Con el uso de este patrón las clases controladoras se encargan de ejecutar acciones de acuerdo a las peticiones que le

llegan y las de acceso a datos interactúan con las tablas de la base de datos directamente, de forma tal que se elimina la sobrecarga de funcionalidades en las clases controladoras. Una clase implementada posee un número relativamente pequeño de responsabilidades, con una importante funcionalidad relacionada y poco trabajo por hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande. Implementando este patrón mejora la calidad y facilidad con que se puede entender el diseño, se genera un bajo acoplamiento y permite fomentar la reutilización.

Con la utilización de una arquitectura N_Capas se contribuye a mantener una Alta Cohesión y un Bajo Acoplamiento ya que las clases no se encuentran saturadas de responsabilidades, además, presentan poca dependencia entre sí, de forma que en caso de realizar cambios en la Capa de Negocio se tenga la mínima repercusión posible en la Capa Acceso_Datos.

Patrón Controlador: Responsable de gestionar un evento de entrada al sistema.
El Patrón Controlador es un intermediario entre la Capa de Presentación y el núcleo de las clases donde reside la lógica de la aplicación. Este patrón no realiza mucho trabajo por sí mismo; más bien es el encargado de coordinar la actividad de otros objetos. Se hace uso del Patrón Controlador definiendo clases controladoras que dirigen todo el flujo de información. Por ejemplo, la clase N_GestionarInvestigador es la encargada de manejar todo el flujo de información entre la presentación y el acceso a datos, como se muestra en la Figura 7.

```
public class N_GestionarInvestigador
{

    public N_GestionarInvestigador()
    {

    }

    public void Crear_Investigador( Boolean esAdministrador, Boolean esInvestigador,
            Date fechaCreacion, Boolean estado, String nombre, String primerApellido,
            String segundoApellido,String ci, String usuario, String contrasena)
    {
        String pass = N_Codificar.md5(contrasena);
        A_GestionarInvestigador acceso=new A_GestionarInvestigador();
        Investigador investigador= new Investigador(esAdministrador,
                  esInvestigador, fechaCreacion, estado, nombre, primerApellido,
                  segundoApellido, ci, usuario, pass);
        acceso.Insertar_Investigador(investigador);

    }
```

Figura 7. Patrón Controlador

2.4. Diagrama de clases

Un diagrama de clases se implementa para visualizar las relaciones entre las clases que involucra el sistema, las cuales pueden ser asociativas, de herencia, de uso y de agregación, ya que una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, métodos, relaciones y

semántica; mostrando un conjunto de elementos que son estáticos, como las clases y tipos junto con sus contenidos y relaciones.

A continuación se muestra el diagrama de clases para el escenario Generar Reporte, los restantes diagramas de clases pertenecientes al sistema se pueden apreciar en el **Anexo III**.

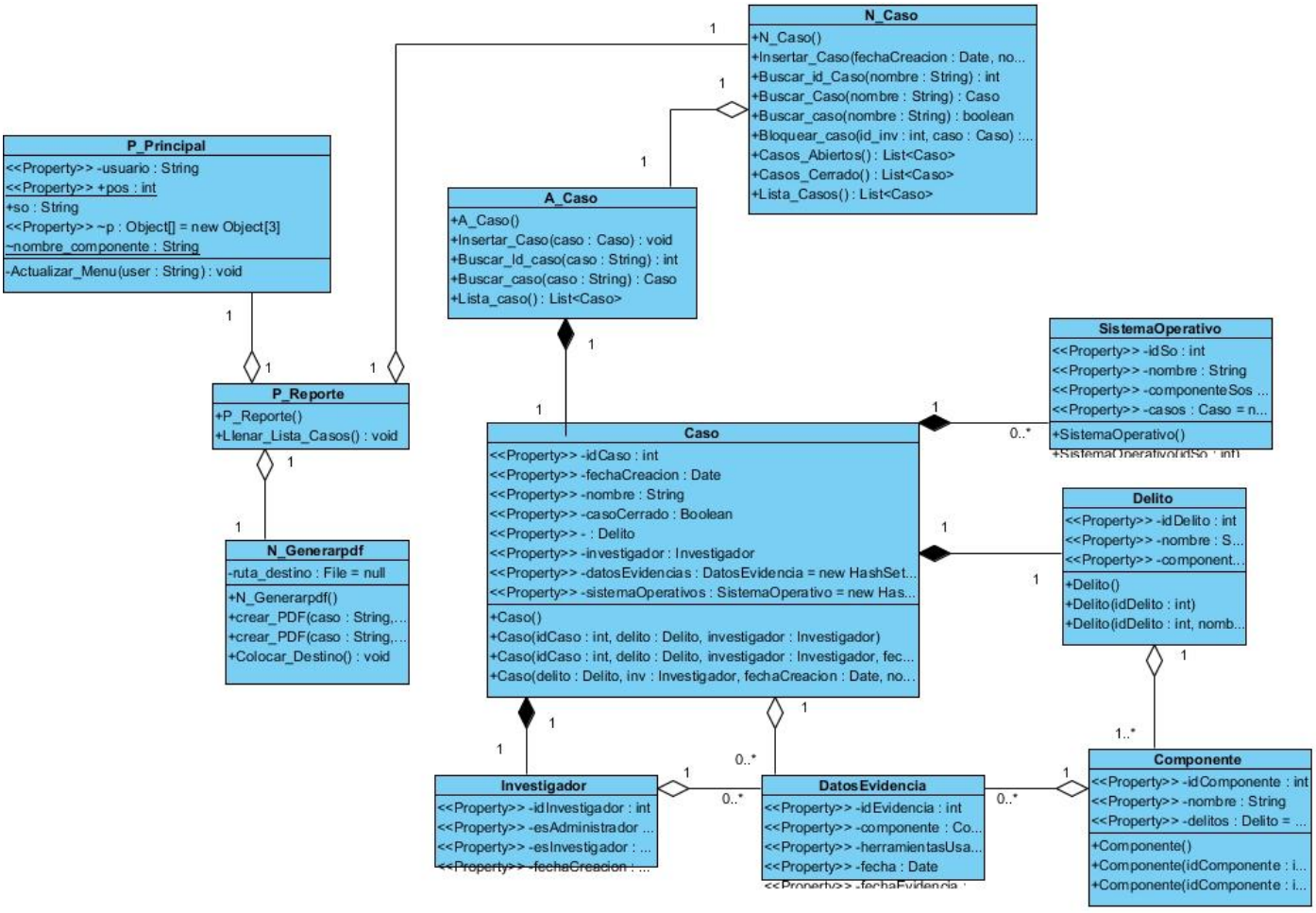


Figura 8. Diagrama de clases-"Generar Reporte"

2.5. Diagrama de la base de datos

En la Figura 9 se expone gráficamente la estructura de la base de datos que se define para el sistema de registro del análisis de evidencias para un caso de informática.

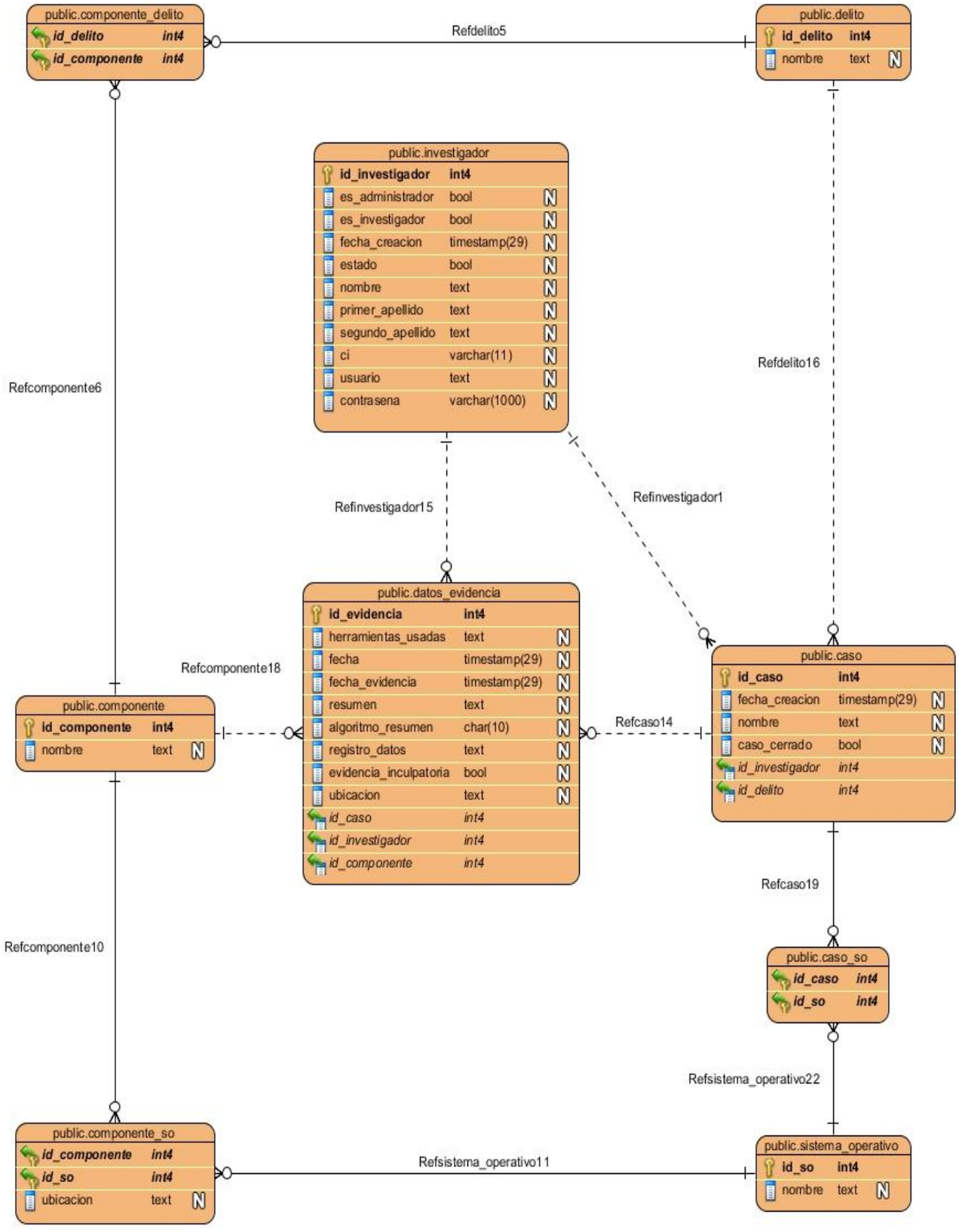


Figura 9. Diseño de la base de datos

Conclusiones

La utilización de la metodología MSF Ágil permitió realizar la descripción del producto a desarrollar, la definición de los escenarios, su priorización y descripción para un mayor entendimiento del proceso y el levantamiento de los requisitos de calidad de servicios; para desarrollar una aplicación ajustada a los principios definidos.

Con la utilización del estilo arquitectónico por capas se puede tratar cada capa del sistema de registro del análisis de evidencia como un sistema independiente.

En este capítulo se ha especificado el comportamiento del sistema a través de los diagramas de clases y el modelado de la base de datos para lograr una mejor comprensión de lo que se está construyendo, obteniendo una visión general con la ayuda de UML.

Capítulo 3. Implementación y prueba

Introducción

En este capítulo se incluyen el diagrama de Componentes y de Despliegue, así como las pruebas pertinentes, logrando así ampliar el abanico de detección de errores en el sistema y poder corregirlos antes de desplegarlo, consiguiendo verificar que el sistema se comporte según los requerimientos establecidos. Se describen los estilos de códigos utilizados con el propósito de lograr un mayor entendimiento del código.

3.1. Diagrama de componentes

Para lograr una organización lógica de la implementación de un sistema se hace necesario el diagrama de Componentes "el cual se conforma de componentes y dependencias entre ellos" (Rumbaugh, 2007). La descripción de los elementos físicos que describen el sistema, así como las relaciones entre ellos, son expuestos en el diagrama de la Figura 10.

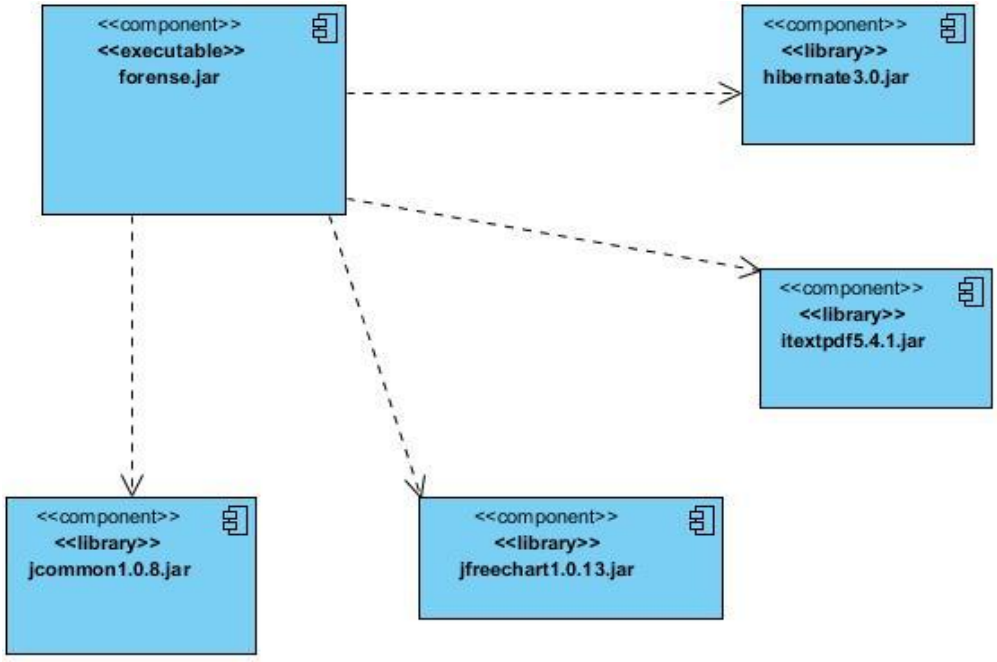


Figura 10. Diagrama de Componentes

Los componentes indicados en el sistema son los necesarios para que la aplicación de escritorio funcione correctamente:

- forense.jar: Es una aplicación de escritorio que para su correcta ejecución depende de los siguientes componentes:
- hibernate 3.0.jar: Contiene las herramientas necesarias para el trabajo con el ORM Hibernate, necesario para la persistencia de los datos por su interacción con el sistema gestor de bases de datos.
- itextpdf 5.4.1.jar: Esta librería está programada para la creación de ficheros 'pdf', es una dependencia porque es indispensable para la generación de reportes en la aplicación.
- jfreechart 1.0.13.jar y jcommon 1.0.8.jar: Permiten la creación de gráficas. Sin estas librerías no se puede mostrar la línea de tiempo.

3.2. Diagrama de despliegue

Es necesario modelar para concebir el sistema, para ello se hace uso del diagrama de Despliegue “el cual se conforma de nodos de procesamiento y componentes para una correcta configuración del sistema en tiempo de ejecución" (Rumbaugh, 2007).

El diagrama de despliegue brinda una visión clara de cuántos nodos participan en el sistema, en este caso se tienen dos nodos, como se muestra en la Figura 11.

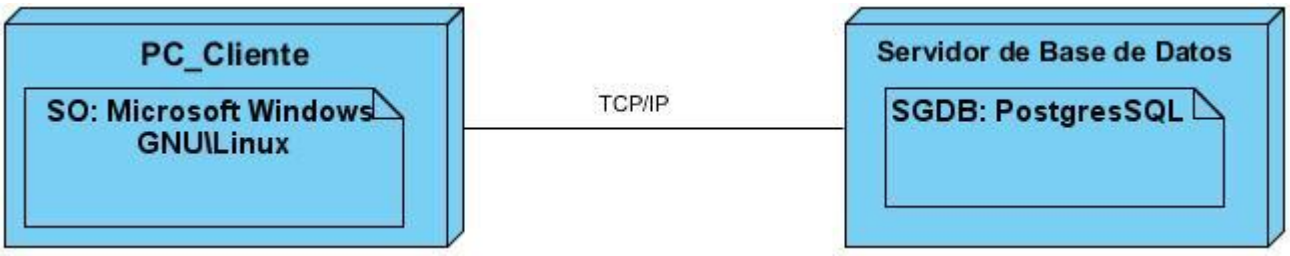


Figura 11. Diagrama de Despliegue

PC-Cliente: Es el nodo donde la aplicación de escritorio será instalada, no existe diferencia cuando se tienen varias computadoras con la misma aplicación.

Servidor de Base de Datos: Es el nodo donde la base de datos estará ejecutándose, cada nodo cliente accederá a la información a través del protocolo TCP/IP.

3.3. Estilo de código

Un estándar de codificación es un conjunto de directrices, normas y reglamentos sobre la forma de escribir el código de un programa, cuyo objetivo fundamental es lograr un mayor entendimiento entre todas las personas que trabajan directamente sobre el código. La coherencia y uniformidad lograda a través del uso de los estándares de codificación constituyen un factor clave para lograr el éxito en el mantenimiento de programas.

El lenguaje de programación seleccionado para el desarrollo del sistema es Java, a continuación se detallan las pautas seguidas para la implementación de la solución.

Principios generales de nomenclatura

Los nombres de cada uno de los elementos del programa, clase o interfaz deben ser significativos, su nombre debe explicar, siempre que sea posible, el uso o fin del elemento; se deben nombrar usando sustantivos (posiblemente compuestos) o formas verbales en imperativo. La forma de construir los nombres será colocando primero el verbo o sustantivo, seguido de cada uno de sus complementos con la primera letra en mayúscula. El idioma seleccionado para realizar la codificación será el español, únicamente cuando se tengan nombres o palabras claves en el negocio cuya traducción resulte engorrosa o poco intuitiva, se procederá a escribirlas en idioma inglés. A continuación, en la Tabla 10 se describen las reglas fundamentales:

Tipos de identificadores	**Reglas para nombres**	**Ejemplos**
Clases	Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Se debe intentar mantener los nombres de las clases descriptivos y simples. Evitar acrónimos y abreviaturas. Cuando la clase está precedida por la letra A, N y P indica que pertenece a la capa Acceso_Datos, Negocio y Presentación respectivamente.	A_Datos_evidencia N_Gestionar_investigador P_Abrir_caso
Interfaces	Los nombres de las interfaces siguen la misma regla que las clases.	P_Registrar_evidencia P_Insertar_investigador P_Guardar_caso
Métodos	Los nombres de los métodos deben ser verbos intuitivos que indiquen lo que se quiere hacer, además empezarán siempre con minúscula. Cuando son compuestos, tendrán la primera letra del nombre en minúsculas, y la primera letra de las palabras siguientes en mayúsculas.	buscarComponente crearInvestigador listarDelitos

Variables	Todas las variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en minúscula igual y mayúsculas las que le siguen. Los nombres de variables no deben empezar con los caracteres guión bajo "_" o signo de peso "$", aunque ambos están permitidos por el lenguaje. Los nombres de las variables deben ser cortos pero con significado. Se deben evitar los nombres de variables de un solo carácter, excepto para variables índices temporales.	String herramientas Int dia boolean evidencia

Tabla 10. Principios de nomenclatura

3.4. Pruebas unitarias

En la programación, una prueba unitaria o de unidad es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada una de las partes que integran la aplicación funcione correctamente por separado (Carlos, 2011). Las pruebas unitarias se realizan para controlar el funcionamiento de pequeñas porciones de código como son rutinas (en la programación estructurada) o métodos (en la programación orientada a objeto). Generalmente son realizadas por los mismos programadores puesto que al conocer con mayor detalle el código, se les simplifica la tarea de elaborar conjuntos de datos de prueba para probarlo. (Suárez, 2003)

La Tabla 11 muestra la prueba unitaria realizada con la herramienta JUnit, a una de las funcionalidades más importantes.

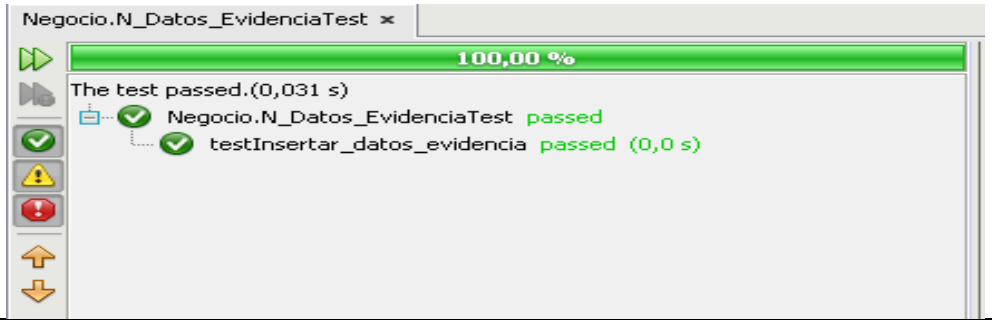
Prueba de Unidad		
Nombre Prueba: N_Datos_EvidenciaTest		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Última Ejecución:**10/5/2013
Ejecutado por: Yanelis Avila Escuela		**Verificador por:** Alina Surós Vicente
Descripción: Es el responsable de insertar los datos de la evidencia en la base de datos.		
Entrada: Componente componente, Investigador investigador, Caso caso, String herramientasUsadas, Date fecha, Date fechaEvidencia, String algoritmoResumen, String registroDatos, Boolean evidenciaInculpatoria, String ubicacion		
Criterio de aceptación: Los datos de la evidencia han sido insertados en la base de datos		
Resultado: 		

Tabla 11. Descripción de la prueba unitaria N_DatosEvidenciaTest

El ícono verde que se ubica al lado del nombre del *test* indica que se completó con éxito, en caso de haber fallado el color que se muestra es rojo. La barra de progreso en verde asegura que la totalidad de las pruebas se ejecutaron satisfactoriamente.

3.5. Pruebas de caja negra

Las pruebas de caja negra son las que se llevan a cabo sobre la interfaz del *software*, por lo que los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene (Pressman, 2001).

A continuación se muestra la Tabla 12 con el caso de prueba realizado al escenario **Crear Reporte**, los restantes se podrán ver en el **Anexo IV**.

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar la opción "Reporte"	Permitir Seleccionar opción "Línea de tiempo", "Reporte del Caso"	NA Reporte	Muestra un menú desplegable y permite seleccionar la opción "Línea de tiempo", "Reporte del Caso"	Opción Reporte
EC 1.2 Seleccionar la opción "Reporte del caso"	Permitir Seleccionar opción "Reporte del caso"	NA Reporte_C	Muestra un menú desplegable y permite seleccionar la opción "Reporte del Caso"	Opción Reporte/ Reporte del caso

EC 1.3 Seleccionar los datos del caso	Permitir seleccionar los datos del caso al que se va a generar el reporte	NA Datos_C	Se marca en azul el caso seleccionado	Opción Reporte/ Reporte del caso/Seleccionar un caso
		I Datos_C	Muestra un mensaje de error : "Debe seleccionar un caso para guardar el reporte"	
EC 1.3 Seleccionar la opción "Crear reporte"	Permitir crear el reporte del caso seleccionado	NA C_Reporte	Permite crear el reporte del caso seleccionado, permite seleccionar dónde lo desea guardar	Opción Reporte/ Reporte del caso/Crear reporte
EC 1.4 Seleccionar la opción "Cancelar"	Permitir cancelar la opción “Crear reporte”	NA Cancelar	Permite cancelar la opción “Crear reporte”	Opción Cancelar

Tabla 12. Descripción del caso de prueba del escenario “Crear Reporte”

Resultado de las pruebas de caja negra

Las pruebas de caja negra fueron aplicadas sobre una versión estable del producto, realizándose tres iteraciones a cada uno de los requisitos del sistema. A continuación en la Tabla 13 se muestra el resultado de aplicar las pruebas de caja negra, en la primera iteración se analizaron tres de diez escenarios en total detectándose 15 no conformidades siendo todas resueltas. En la segunda iteración se analizaron tres de los siete escenarios restantes detectándose 11 no conformidades siendo todas resueltas. En la tercera iteración se analizaron los cuatro escenarios restantes detectándose 20 no conformidades siendo todas resueltas. Para un total de tres iteraciones realizadas, diez escenarios analizados, 46 no conformidades detectadas y 46 no conformidades resueltas.

Iteración	Escenarios	NC_Detectadas	NC_Resueltas
1	3	15	15
2	3	11	11
3	4	20	20

Tabla 13. Resultados de las pruebas de caja negra

En la Figura 12 se muestran de forma gráfica los resultados obtenidos anteriormente:

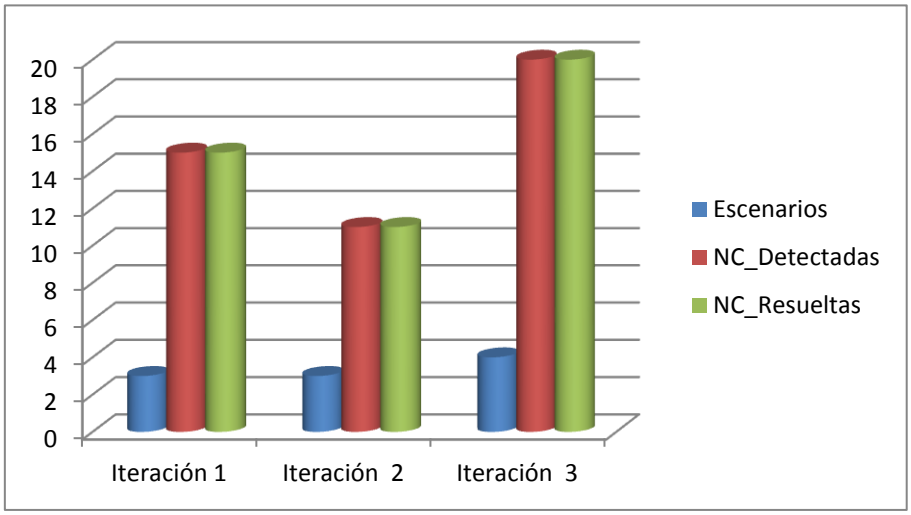


Figura 12. Gráfica del resultado de las pruebas de caja negra

Al concluir la tercera iteración las pruebas de caja negra permitieron comprobar que el sistema de registro del análisis de evidencia cumple con los requerimientos.

Conclusiones

Se especificaron los diagramas de Componentes y de Despliegue para una mayor comprensión del sistema en general.

Las pruebas de caja negra y unitarias realizadas posibilitaron evaluar la calidad del producto, desde las respuestas de la aplicación ante las entradas de los datos hasta el comportamiento de las clases y métodos de la aplicación.

Conclusiones generales

El estudio del estado del arte sobre análisis de evidencia permitió evaluar los criterios de admisibilidad, manipulación y gestión de la evidencia digital, concluyéndose la utilización del modelo de formalización de la evidencia digital en la solución propuesta.

Con el estudio de sistemas para el análisis forense digital, se demuestra que existen soluciones libres para el análisis de evidencias digitales así como propietarias, pero que tienen como desventaja no asistir al investigador con los elementos específicos de la evidencia.

El análisis de las diferentes tecnologías, metodologías, lenguajes y herramientas según las necesidades de la solución, hizo posible la selección adecuada para el desarrollo del sistema.

Se desarrolló un sistema que cumple con las funcionalidades necesarias para realizar el registro del análisis de evidencia para un caso de informática forense, apoyándose en la especificación de escenarios, el correcto diseño de la arquitectura y uso de los patrones de diseño.

Las pruebas de caja negra y unitarias realizadas posibilitaron evaluar la calidad del producto, desde las respuestas de la aplicación ante las entradas de los datos hasta el comportamiento de las clases y métodos de la aplicación.

Recomendaciones

Desplegar el sistema en la dirección de seguridad informática de la UCI.

Bibliografía referenciada

Accesdata, Accesdata. 2012. Accesdata Accesdata. [En línea] 2012. [Citado el: 4 de Febrero de 2012.] http://www.accessdata.com/products/digital-forensics/ftk..

ACPO. 1999. *Good practice guide for computer based electronic evidence.* 1999.

Acurio, Santiago. 2007. *Introducción a la informática forense.* 2007.

Alonso Velázquez, José Luis. 2010. *Lenguajes de Programación.* Universidad de Guanajuato : s.n., 2010.

Análisis forense en computadoras. **Zoratto, Carlos Alberto. 2001.** 2001.

Archival Application of Digital Forensics Methods for Authenticity, Description and Access Provision. **Lee, Cal. 2012.** Australia : s.n., 2012. International Council on Archives Congress.

Arquillo Cruz, José. 2007. *Herramienta de apoyo para el análisis forense de computadoras.* s.l. : Universidad de Jaén, 2007.

Baggili, Ibrahim. 2011. *Digital forensics and Cyber Crime.* New York : ICST, 2011. ISBN: 978-3-642-19512-9.

Brezinski, D.,Killalea, T. 2002. *RFC 3227: Guidelines for Evidence Collection and Archiving. Network Working Group.* 2002.

Callao, Universidad Nacional del. 2008. *Teoría de lenguaje de programación.* 2008.

Camacho, Erika. 2004. *Arquitecturas de Software.* 2004.

Cano Martínez, Jeimy Jose. 2003. *Admisibilidad de la Evidencia Digital: Algunos Elementos de Revisión y Análisis.* 2003.

—. 2006. *Introducción a la Informática Forense .* 2006.

Carlos, B. 2011. *QUnit, testeando nuestras aplicaciones.* 2011.

Carrier, Brian. 2005. *File System Forensic Analysis.* s.l. : Addison Wesley Professional, 2005. ISBN: 0-32-126817-2.

Casey, Eoghan. 2004. *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet.* 2004.

Certain Jaramillo, Andrés Felipe, Mosquera González, José Alejandro y Cano Martínez, Jeimy José. 2005. *Evidencia Digital: contexto, situación e implicaciones nacionales.* 2005.

Deftlinux. 2013. Deftlinux. [En línea] 2013. [Citado el: 4 de Febrero de 2013.] http://www.deftlinux.net..

Distintas perspectivas de los delitos informáticos . **Ferreyro, Lorena B. 2005.** Buenos Aires : s.n., 2005.

Dittrich, David. 2002. *Análisis Forense de Sistemas GNU/Linux, Unix.* 2002.

—. 2013. *Basic Steps in Forensic Analysis of Unix Systems.* 2013.

Ecured. 2012. Enciclopedia colaborativa en la informatización cubana. [En línea] 2012. [Citado el: 18 de Marzo de 2013.] www.ecured.cu/index.php/Mapeo_de_objetos.

El delito informático. **Hernández Días, Leyre. 2009.** 23, San Sebastián : s.n., 2009.

Embarcadero Technologies Inc. 2009. Embarcadero. [En línea] 2009. [Citado el: Marzo de 15 de 2013.] www.embarcadero.com/products/er-studio.

Estudioteca. 2013. Estudioteca. [En línea] 2013. [Citado el: 16 de Marzo de 2013.] http://www.estudioteca.net/universidad/telecomunicaciones/gestor-base-datos.

Extremeprograming. 2009. Extremeprograming. [En línea] 2009. [Citado el: 18 de Marzo de 2013.] http://www. extremeprograming.org.

Gamma, Erich, Helm, Richard, Johnson, Ralph, Vlissides, John. 2003. *Patrones de Diseño.* Madrid : Addison Wesley, 2003. ISBN 84-7829-059-1.

García Martínez, Antonio Javier. 2001. *La formación de un IRT(Incident Response Team) forense.* 2001.

Garfinkel, Simson L. 2011. *Digital forensics research: The next 10 years.* s.l. : Elsevier, 2011.

Ghosh, Ajoy. 2004. *Guidelines for the Management of IT Evidence.* 2004.

Guidance Software. 2012. Guidance Software. [En línea] 2012. [Citado el: 4 de Febrero de 2013.] http://www.guidancesoftware.com.

IOCE. 2013. Organización Internacional de Evidencia Informática. [En línea] 2013. [Citado el: 3 de Marzo de 2013.] http://www.ioce.org.

ISFS. 2004. *Computer forensics: Best Practices.* 2004. Vol. II.

Jiménez Ruiz, Beymar, Peña Pérez, Sandra. 2012. *Metodología de desarrollo de software-MSF.* 2012.

Justice National Institute. 2004. *Forensic examination of digital evidence. A guide for law enforcement .* 2004.

Kirschenbaum, Matthew G. Ovenden, Richard, Redwine, Gabriela. 2010. *Digital Forensics and Born-Digital Content in Cultural Heritage Collections.* 2010.

Larman, Craig. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : s.n., 1999. ISBN: 970-17-0261-1.

Leigland, Ryan. 2004. *A Formalization of Digital Forensics.* s.l. : International Journal of Digital Evidence Fall, 2004.

León, Ricardo, Haver, Amaya, López, Oscar. 2001. *Informática forense: generalidades, aspectos técnicos y herramientas.* 2001.

Littlejohn Shinder, Debra. 2002. *Scene of the cybercrime . Computer Forensics Handbook .* s.l. : Syngress Publishing, Inc., 2002. ISBN: 1-931836-65-5.

López Delgado, Miguel. 2007. *Análisis Forense Digital.* 2007.

Mandia, Kevin. 2001. *Initial Response to Windows.* s.l. : Foundstone, 2001.

Marcella, Albert J. y Greendfield, Robert S. 2001. *Cyber Forensics. A field manual for Collecting, Examining, and Preserving Evidence of Computer Crimes.* s.l. : Auerbach Publications, 2001. ISBN 0−8493−0955−7.

Martínez, Rafael. 2010. Postgresql. [En línea] 2010. [Citado el: 16 de Marzo de 2013.] http://www.postgresql.org.es/sobre_postgresql.

Microsoft. 2007. MSF for Agile Software Development. [En línea] 2007. [Citado el: 18 de Marzo de 2013.] www.microsoft.com.

Middleton, Bruce. 2002. *Cyber Crime Investigator's Field Guide.* s.l. : Auerbach Publications, 2002. ISBN 0-8493-1192-6.

Mohay, George, y otros. 2003. *Computer and Intrusion Forensics.* s.l. : Artech House, 2003. ISBN 1-58053-369-8.

Montero Garrido, Jesús Manuel. 2001. *Plataforma Eclipse.* 2001.

Mukasey, Michael B, Sedgwick, Jeffrey L., Hagy, David W. 2008. *Electronic Crime Scene Investigation: A Guide for First Responders.* 2008. Vol. II.

Netbeans, Oracle Corporation. 2013. NetBeans. [En línea] 2013. [Citado el: 9 de Marzo de 2013.] https://netbeans.org..

Network, Microsoft Developer. 2013. Network Microsoft Developer. [En línea] 2013. [Citado el: 20 de Marzo de 2013.] http://msdn.microsoft.com/.

Noblett, Michael G. 2000. *Recovering and Examining Computer Forensic Evidence.* 2000.

Paradigm, Visual. 2011. Visual Paradigm. [En línea] 2011. [Citado el: 15 de Marzo de 2013.] http://www.visual-paradigm.com.

Pladna, Brett. 2010. *Computer Forensics Procedures, Tools, and Digital Evidence Bags: What They Are and Who Should Use Them.* 2010.

Presman, Gustavo Daniel. 2010. *Introducción al análisis forense de dispositivos móviles.* 2010.

Pressman, Roger S. 2001. *Ingeniería del Software.* Madrid : Mc Graw Hill, 2001. ISBN 8448132149.

Rumbaugh, James, Jacobson, Ivar, Grady Booch. 2007. *El Lenguaje Unificado de Modelado. Manual de referencia.* Madrid : Booch Pearson Addison Wesley, 2007. ISBN 978-84-782-9074-1.

Schatz, Bradley. 2007. *Digital Evidence:Representation and Assurance.* 2007.

Schildt, Herbert. 2001. *Java 2. Manual de referencia.* Madrid : McGraw-Hill, 2001. ISBN: 84-481-3173-8.

Selective and intelligent imaging using digital evidence bags. **Turner, Philip. 2006.** s.l. : Elsevier, 2006.

Sleuthkits. 2013. Sleuthkits. [En línea] 2013. [Citado el: 4 de Febrero de 2013.] http://www.sleuthkit.org/autopsy/desc.php.

Solís, Luis alberto. 2012. *Análisis forense en dispositivos Android.* Ecuador : s.n., 2012.

Suárez, Pablo, Fontela, Carlos. 2003. *Documentación y pruebas antes del paradigma de objetos.* 2003.

Tellez Valdéz, Julio. 2008. *Derecho Informático.* México : s.n., 2008. Vol. II.

Toledo Alma, Enríquez y Ayala, Jesús Maldonado. 2013. UAEM. [En línea] 2013. [Citado el: 16 de Marzo de 2013.] www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf.

Usaola, Macario Polo. 2006. Pruebas de programas Java mediante JUnit. [En línea] 2006. [Citado el: 19 de Marzo de 2013.] http://www.inf-cr.uclm.es/www/mpolo/tutorial/.

Valenzuela Espejo, Ismael. 2008. *Análisis de herramientas en Sistemas Windows.* 2008.

Bibliografía consultada

Bem Derek and Huebner Ewa Computer Forensic Analysis in a Virtual Environment [Journal]. - Australia : International Journal of Digital Evidence, 2007. - 2 : Vol. VI.

Bunting Steve Encase Computer Forensics [Book]. - [s.l.] : Wiley Publishing, 2008.

Carrier Brian File System Forensic Analysis [Book]. - [s.l.] : Addison Wesley Professional, 2005.

Cohen Fred Challenges to Digital Forensic Evidence [Journal]. - 2006.

Cohen Michael PyFlag – An advanced network forensic framework [Journal]. - [s.l.] : Elsevier, 2008.

Ferreyro Lorena B. Distintas perspectivas de los delitos informáticos [Conference] // Conferencia Internacional sobre seguridad informática. - Argentina : [s.n.], 2005.

Hernández Díaz Leyre El delito Informático [Journal]. - 2009.

Hosmer Chet Proving the Integrity of Digital Evidence with Time [Journal]. - [s.l.] : International Journal of Digital Evidence, 2002. - 1 : Vol. I.

Lee Cal Archival Application of Digital Forensics Methods for Authenticity, Description and Access Provision [Journal]. - Australia : [s.n.], 2012.

Littlejohn Shinder Debra Scene of the Cybercrime. Computer Forensics Handbook [Book]. - [s.l.] : Syngress Publishing, 2002.

Marcella Albert J. and Greendfield Robert S. Cyber Forensics. A field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes [Book]. - [s.l.] : Auerbach Publications, 2002.

Masters Gerry and Turner Philip Forensic data recovery and examination of magnetic swipe card cloning devices [Journal]. - [s.l.] : Elsevier, 2007.

Middleton Bruce Cyber Crime. Investigator´s Field Guide [Book]. - [s.l.] : Auerbach Publications, 2002.

Mohay George [et al.] Computer and intrusion forensics [Book]. - [s.l.] : Artech House, 2003.

Mukasey Michael B., Sedgwick Jeffrey L. y Hagy David W. Electronic Crime Scene Investigation: A Guide for First Responders, Second Edition [Publicación periódica]. - 2008.

Ossa Rojas Claudio Patricio Apreciaciones generales sobre los delitos informáticos [Journal].

Pladna Brett Computer Forensics Procedures, Tools, and Digital Evidence Bags: What They Are and Who Should Use Them [Journal].

Richard Golden G., Roussev Vassil and Marziale Lodovico Forensic discovery auditing of digital evidence containers [Journal]. - [s.l.] : Elsevier, 2007.

Rogers Marcus K. Cyber Forensics: Evidence Collection, Management and Handling [Journal]. - 2009.

Schatz Bradley and Clark Andrew An open architecture for digital evidence integration [Journal]. - 2006.

Turner Philip Selective and intelligent imaging using digital evidence bags [Journal]. - [s.l.] : Elsevier, 2006.

Zoratto Carlos Alberto Análisis forense de computadoras [Journal]. - Argentina : [s.n.].

Glosario de términos

B

Booteable: Puede arrancar directamente desde el dispositivo en que está grabado el programa sin intervención del sistema operativo que tenga instalado el ordenador.

D

Disco duro IDE: Es una unidad que usa una conexión ATA paralela, llamada IDE de manera informal. El término IDE es sinónimo de *Integrated Drive Electronics Interface* (Interfaz Electrónica de Unidad Integrada). La primera vez que los discos duros usaron la interfaz IDE fue en 1986.

H

Hash: Algoritmo para generar claves para identificar de manera única a un documento, registro o archivo.

Hex: El actual formato 'hex' fue ideado por Intel para la descripción del código objeto de sus microprocesadores de 8, 16 y 32 bits. Es un formato flexible porque toda la información se representa en ASCII lo que ofrece muchas prestaciones.

I

Instancia: Se produce con la creación de un objeto perteneciente a una clase. El objeto que se crea tiene los atributos, propiedades y métodos de la clase a la que pertenece. Los objetos y sus características se usan en la construcción de programas, ya sea como contenedores de datos o como partes funcionales del programa. Los objetos también puede ser ocurrencia de las clases.

L

Lenguaje de programación: Es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático.

Live CD: Es un sistema operativo almacenado en un CD, que puede ser ejecutado desde éste sin que haya necesidad de instalarlo en el disco duro.

Log: Archivo que registra movimientos y actividades de un determinado programa, usuario o proceso.

M

Malware: Es una variedad de *software* que tiene como finalidad infiltrarse, dañar o causar un mal funcionamiento en una computadora sin el consentimiento de su propietario, los virus y gusanos son ejemplos de este tipo de *software*.

MD5: Es un algoritmo de reducción criptográfico que fue desarrollado por Ronald Rivest en 1995 y está basado en dos algoritmos anteriores MD2 y MD4, este algoritmo genera un número de 128 bits basado en el contenido de un fichero que ha sido publicado en la web.

Metadatos: Son datos que describen o definen algún aspecto de un recurso de información (como un documento, una imagen, una página web, etc.).

N

NTFS: *New Technology File System* (Nueva Tecnología De Sistema De Archivos), es el estándar de sistema de archivos usada por Microsoft Windows sustituye al FAT.

P

Programación estructurada: Es un paradigma de programación orientado a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora, utilizando únicamente subrutinas y tres estructuras: secuencia, selección (*if* y *switch*) e iteración (bucles *for* y *while*).

R

RAW: "Formato de Imagen sin modificaciones" es un formato de archivo digital de imágenes que contiene la totalidad de los datos de la imagen tal y como ha sido captada por el sensor digital de la cámara fotográfica.

S

SCSI: *Small Computers System Interface* (Sistema de Interfaz para pequeñas computadoras), interfaz estándar para la transferencia de datos entre distintos dispositivos de la computadora.

Software: *Software* o programa, es un conjunto de componentes lógicos necesarios que hacen posible la realización de tareas en una computadora.

SWAP: Espacio de intercambio de un disco.

T

TCP/IP: Define las reglas que deben seguir las computadoras para comunicarse con otras en una red, pueden ser interna (red local o intranet) o en la red global (internet).

V

Virus: Un virus informático es un *software* malicioso que necesita de la intervención del hombre para propagarse, se adjunta a un programa o archivo para replicarse a sí mismo y continuar infectando el ordenador.

Anexos

Anexo I. Descripción de Delito-SO-Componente-Ubicación.

Delito	SO	Componentes	Ubicación	Bibliografía
Intrusión en una computadora	MAC	Libreta de direcciones	~/Library/Addresses	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Archivos de configuración	/etc ~/Library/Preferences	
		Programas ejecutables	/usr/bin /sw/bin /Applications	
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
		Historial del navegador/cache	~/Library/Caches/MS/Internet/Cache ~/Library/Caches Safari	
		Dirección IP y nombre de usuario	System Preferences/Network/ ~/Library	
		Archivos temporales	/tmp/	
	Linux	Ficheros de configuración	Sistema: /etc Usuario: ~/	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Correo electrónico	~/.thunderbird	
		Programas ejecutables	/sbin /usr/bin /usr/local/bin	
		Log de actividad en internet	~/.mozilla	
		Historial del navegador/cache	~/.mozilla/firefox/CRYPT/Cache	
		Dirección IP y nombre de usuario	/etc/network/interfaces /etc/passwd	
		Código Fuente	/usr/src/	
	Windows	Correo electrónico	C:\Users\Nombre de usuario \Documents\Archivos de Outlook	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (García Martínez,
		Libreta de direcciones	C:\Documents and Settings\Usuario \Datos de programa\Microsoft\Address Book	
		Programas ejecutables	Inicio\buscar\	
		Log de actividad en internet	shell:Cache	

		Historial del navegador/cache	Mozilla: offlineCache. Datos para trabajar sin conexión cookies.sqlite. Contiene las cookies formhistory.sqlite. El historial downloads.sqlite. Historial de descargas signons.sqlite. Contraseñas guardadas	2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Dirección IP y nombre de usuario	Conexiones de red	
		Log de chat	C:\Documents and Settings\Nombre de usuario\Datos de programa\Pandion	
	Otros	Libreta de direcciones		
		Ficheros de configuración		
		Correo electrónico		
		Programas ejecutables		
		Log de actividad en internet		
		Historial del navegador/cache		
		Dirección IP y nombre de usuario		
		Código fuente		
		Fichero de texto		
Investigación de muerte	MAC	Libreta de direcciones	~/Library/Addresses	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Diario	~/Documents	
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Imágenes	~/Images	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
		Historial del navegador/cache	~/Library/Caches/MS/ Internet/ Cache ~/Library/Caches Safari	
	Linux	Correo electrónico	~/.thunderbird	(ACPO, 1999) (Casey, 2004) (Leigland, 2004)
		Registro de bienes/financieros	/home/Nombre de Usuario/Imágenes	

		Imágenes	/home/Nombre de Usuario/Imágenes	(Mukasey, 2008)
		Log de actividad en internet	~/.mozilla	
		Historial del navegador/cache	~/.mozilla/firefox/CRYPT/Cache	
		Documentos legales y testamentos	/home/Nombre de usuario/Documentos	
		Registros médicos	/home/Nombre de usuario/Documentos	
	Windows	Correo electrónico	C:\Users\Nombre de usuario\Documents\Archivos de Outlook	(ACPO, 1999) (Casey, 2004) (Leigland, 2004) (García Martínez, 2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Libreta de direcciones	C:\Documents and Settings\Usuario \Datos de programa\Microsoft\Address Book	
		Registro de bienes/financieros	C:\Documents and Settings\Nombre de usuario\Mis documentos	
		Imágenes	C:\Documents and Settings\Nombre de usuario \Mis documentos\Mis imágenes	
		Log de actividad en internet	shell:Cache	
		Historial del navegador/cache	C:\Documents and Settings\Nombre de usuario\Configuración local\Archivos temporales de Internet	
		Documentos legales y testamentos	C:\Documents and Settings\Nombre de usuario\Mis documentos	
		Registros médicos	C:\Documents and Settings\Nombre de usuario\Mis documentos	
	Otros	Libreta de direcciones		
		Diario		
		Correo electrónico		
		Registro de bienes/financieros		
		Imágenes		
		Log de actividad en internet		
		Historial del navegador/cache		
		Documentos legales y testamentos		
		Registros médicos		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007) (García Martínez, 2001)
		Calendario	/data/data/com.android.providers.calendar/	
		SMS&MMS	/data/data/com.android.providers.telephon/	

		Historial de descargas	/data/data/com.android.providers.downloas/	(Solís, 2012)
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	
		Cache	/data/data/com.google.android.location/	
Violencia doméstica	MAC	Libreta de direcciones	~/Library/Addresses/	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Diario	~/Documents	
		Correo electrónico	~/Library/Mail/*/*/mbox	
	Linux	Correo electrónico	~/.thunderbird	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Registro de bienes/financieros	/home/Nombre de usuario/Documentos	
		Registros médicos	/home/Nombre de usuario/Documentos	
	Windows	Correo electrónico	C:\Users\Nombredel usuario\Documents\Archivos de Outlook	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (García Martínez, 2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Libreta de direcciones	C:\Documents and Settings\Usuario \Datos de programa\Microsoft\Address Book	
		Registro de bienes/financieros	C:\Documents and Settings\Nombre de usuario \Mis documentos	
		Registros médicos	C:\Documents and Settings\Nombre de usuario \Mis documentos	

	Otros	Libreta de direcciones		
		Diarios		
		Correo electrónico		
		Registro de bienes/financieros		
		Registros médicos		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007)
		Calendario	/data/data/com.android.providers.calendar/	(García Martínez, 2001)
		SMS&MMS	/data/data/com.android.providers.telephon/	(Solís, 2012)
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	
		Cache	/data/data/com.google.android.location/	
Fraude económico	MAC	Libreta de direcciones	~/Library/Addresses	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Calendario	~/Library/Calendars	
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
		Historial del navegador/cache	~/Library/Caches/MS\ Internet\ Cache ~/Library/Caches Safari	
	Linux	Imágenes de cheques/monedas/ giros postales/firmas	/home/Nombre de usuario/imagenes	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Información de comprador/Datos de tarjetas de crédito	/home/Nombre de usuario/Documentos	
		Base de datos	/var/lib/postgresql /var/lib/mysql	
		Correo electrónico	~/.thunderbird	
		Formulario de falsas transacciones financieras	/home/Nombre de usuario/Documentos	
		Registro de bienes/financieros	/home/Nombre de usuario/Documentos	
		Log de actividad en internet	~/.mozilla	
	Windows	Imágenes de	C:\Documents and Settings\Nombre de usuario	

		cheques/monedas/giros postales/firmas	\Mis documentos\Mis imágenes	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (García Martínez, 2001) (Leigland, 2004) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Información de comprador/Datos de tarjetas de crédito	C:\Documents and Settings\Nombre de usuario\Mis documentos	
		Base de datos	C:\Documents and Settings\All Users\Datos de programa\MySQL\MySQL Server \data C:\Archivos de programa\PostgreSQ\data\base	
		Correo electrónico	C:\Users\Nombredel usuario\Documents\Archivos de Outlook	
		Formulario de falsas transacciones financieras	C:\Documents and Settings\Nombre de usuario\Mis documentos	
		Registro de bienes/financieros	C:\Documents and Settings\Nombre de usuario\Mis documentos	
		Log de actividad en internet	shell:Cache	
	Otros	Libreta de direcciones		
		Agendas		
		Imágenes de cheques/monedas/giros postales/firmas		
		Información de comprador/Datos de tarjetas de crédito		
		Base de datos		
		Correo electrónico		
		Formulario de falsas transacciones financieras		
		Registro de bienes/financieros		
		Log de actividad en internet		
		Software de acceso a instituciones financieras online		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007) (García Martínez, 2001) (Solís, 2012)
		Calendario	/data/data/com.android.providers.calendar/	
		SMS&MMS	/data/data/com.android.providers.telephon/	
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	

		Cache	/data/data/com.google.android.location/	
Amenazas/ Acoso por correo electrónico	MAC	Libreta de direcciones	~/Library/Addresses	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Diarios	~/Documents	
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Imágenes	~/Images	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
	Linux	Correo electrónico	~/.thunderbird	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Registro de bienes/financieros	/home/Nombre de usuario/documentos	
		Imágenes	/home/Nombre de usuario/imágenes	
		Log de actividad en internet	~/.mozilla	
		Documentos legales	/home/Nombre de usuario/documentos	
	Windows	Correo electrónico	C:\Users\Nombredel usuario\Documents\Archivos de Outlook	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Libreta de direcciones	C:\Documents and Settings\Usuario \Datos de programa\Microsoft\Address Book	
		Registro de bienes/financieros	C:\Documents and Settings\Nombre de usuario\Mis documentos	
		Imágenes	C:\Documents and Settings\Nombre de usuario\Mis documentos\Mis imagenes	
		Log de actividad en internet	shell:Cache	
		Documentos legales	C:\Documents and Settings\Nombre de usuario\Mis documentos	
	Otros	Libreta de direcciones		
		Diarios		
		Correo electrónico		
		Registro de bienes/financieros		
		Imágenes		
		Log de actividad en internet		

		Documentos legales		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007) (García Martínez, 2001) (Solís, 2012)
		Calendario	/data/data/com.android.providers.calendar/	
		SMS&MMS	/data/data/com.android.providers.telephon/	
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	
		Cache	/data/data/com.google.android.location/	
Extorción	MAC	Marca de tiempo en ficheros	–lR /	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
		Historial del navegador/cache	~/Library/Caches/MS\ Internet\ Cache ~/Library/Caches Safari	
		Archivos temporales	/tmp/	
	Linux	Marca de tiempo en ficheros	stat/nombre del archivo	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Correo electrónico	~/.thunderbird	
		Log histórico	~/.bash_history	
		Log de actividad en internet	~/.mozilla	
		Historial del navegador/cache	~/.mozilla/firefox/CRYPT/Cache	
		Nombre de usuarios	/etc/passwd	
	Windows	Marcas de tiempo de ficheros	Fichero\propiedades\detalles	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (García Martínez, 2001) (Leigland, 2004) (Mukasey, 2008)
		Correo electrónico	C:\Users\Nombredel usuario\Documents\Archivos de Outlook	
		Log de actividad en internet	shell:Cache	
		Nombre de usuarios	C:\Documents and Settings	

	Otros	Marcas de tiempo de ficheros		
		Correo electrónico		
		Log histórico		
		Log de actividad en internet		
		Nombre de usuarios		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007) (García Martínez, 2001) (Solís, 2012)
		Calendario	/data/data/com.android.providers.calendar/	
		SMS&MMS	/data/data/com.android.providers.telephon/	
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	
		Cache	/data/data/com.google.android.location/	
Apuestas ilegales/juegos	MAC	Libreta de direcciones	~/Library/Addresses	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Calendario	~/Library/Calendars	
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Log de Historial	~/.bash_history ~/Library/Safari	
		Imágenes de juegos	/usr/bin /Applications	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
	Linux	Base de datos del cliente y registros del jugador	/var/lib/postgresql /var/lib/mysql	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004)
		Información de comprador/Datos de tarjetas de crédito	/home/Nombre de usuario/documentos	
		Correo electrónico	~/.thunderbird	

				(Mukasey, 2008)
		Registro de bienes/financieros	/home/Nombre de usuario/documentos	
		Log de actividad en internet	~/.mozilla	
		Imágenes de los jugadores	/home/Nombre de usuario/imagenes	
	Windows	Bases de datos del cliente y registros del jugador	C:\Documents and Settings\All Users\Datos de programa\MySQL\MySQL Server \data C:\Archivos de programa\PostgreSQ\data\base	(ACPO, 1999) (Casey, 2004) (Arquillo Cruz, 2007) (García Martínez, 2001) (Leigland, 2004) (Mukasey, 2008)
		Libreta de direcciones	C:\Documents and Settings\Usuario \Datos de programa\Microsoft\Address Book	
		Información de cliente/Datos de tarjeta de crédito	C:\Documents and Settings\Nombre de usuario\Mis documentos	
		Correo electrónico	C:\Users\Nombredel usuario\Documents\Archivos de Outlook	
		Registro de bienes/financieros	C:\Documents and Settings\Nombre de usuario \Mis documentos	
		Log de actividad en internet	shell:Cache	
		Imágenes de los jugadores	C:\Documents and Settings\Nombre de usuario \Mis imágenes	
		Estadísticas de apuestas en deportes	C:\Documents and Settings\Nombre de usuario\Mis documentos	
	Otros	Libreta de direcciones		
		Agendas		
		Bases de datos del cliente y registros del jugador		
		Información de cliente/Datos de tarjeta de crédito		
		Dinero electrónico		
		Correo electrónico		
		Registro de bienenes/financieros		
		Log de actividad en internet		
		Imágenes de los jugadores		
		Software de acceso a instituciones financieras online		
		Estadísticas de apuestas en		

		deportes		
Robo de identidad	MAC	Generadores de tarjetas de crédito	system_profiler Ver generadores de tarjetas de crédito	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Cámaras digitales	system_profiler Ver cámaras digitales	
		Documentos borrados	~/.Trash	
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Escáner	system_profiler ver escáner	
		Ficheros del sistema	~/Library/Logs	
	Linux	Documentos borrados	/dev/sda2	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Escáner	/usr/share/sane/snapscan/	
		Correo electrónico	~/.thunderbird	
	Windows	Lectores/grabadores de tarjetas	Mi PC/Administración de equipos/Administrador de dispositivo	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (García Martínez, 2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Escáner	Mi PC/Administración de equipos/Administrador de dispositivo	
		Correo electrónico	C:\Users\Nombre de usuario \Documents\Archivos de Outlook	
	Otros	Generadores de tarjetas de crédito		
		Lectores/grabadores de tarjetas		
		Cámaras digitales		

		Escáner		
		Correo electrónico		
		Documentos borrados		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007) (García Martínez, 2001) (Solís, 2012)
		Calendario	/data/data/com.android.providers.calendar/	
		SMS&MMS	/data/data/com.android.providers.telephon/	
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	
		Cache	/data/data/com.google.android.location/	
Narcóticos	MAC	Libreta de direcciones	~/Library/Addresses	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Calendario	~/Library/Calendars	
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
	Linux	Base de datos	/var/lib/postgresql /var/lib/mysql	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Formulas/ recetas de drogas	/home/Nombre de usuario/documentos	
		Correo electrónico	~/.thunderbird	
		Registro de bienes/financieros	/home/Nombre de usuario/documentos	
		Log de actividad en internet	~/.mozilla	
		Imágenes/Plantillas de formularios de recetas médicas	/home/Nombre de usuario/imagenes	
	Windows	Bases de datos	C:\Documents and Settings\All Users\Datos de programa\MySQL\MySQL Server \data C:\Archivos de programa\PostgreSQ\data\base	(ACPO, 1999) (Arquillo Cruz, 2007)

		Formulas/ recetas de drogas	C:\Documents and Settings\Nombre de usuario \Mis documentos	(Casey, 2004) (Leigland, 2004) (García Martínez, 2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Correo electrónico	C:\Users\Nombre de usuario \Documents\Archivos de Outlook	
		Registros de bienes/ financieros	C:\Documents and Settings\Nombre de usuario \Mis documentos	
		Log de actividad en internet	shell:Cache	
		Imágenes/Plantillas de formularios de recetas médicas	C:\Documents and Settings\Nombre de usuario \Mis imágenes	
	Otros	Libreta de direcciones		
		Agendas		
		Bases de datos		
		Formulas/ recetas de drogas		
		Correo electrónico		
		Identificación falsa		
		Registros de bienes/ financieros		
		Log de actividad en internet		
		Imágenes/Plantillas de formularios de recetas médicas		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007) (García Martínez, 2001) (Solís, 2012)
		Calendario	/data/data/com.android.providers.calendar/	
		SMS&MMS	/data/data/com.android.providers.telephon/	
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	
		Cache	/data/data/com.google.android.location/	
Prostitución	MAC	Libreta de direcciones	~/Library/Addresses	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004)
		Calendario	~/Library/Calendars	
		Correo electrónico	~/Library/Mail/*/*/mbox	

		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	(Leigland, 2004) (Mukasey, 2008)
	Linux	Base de datos de clientes/ registros	/var/lib/postgresql /var/lib/mysql	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Correo electrónico	~/.thunderbird	
		Registro de bienes/ financieros	/home/Nombre de usuario/documentos	
		Log de actividad en Internet	~/.mozilla	
		Registros médicos	/home/Nombre de usuario/documentos	
	Windows	Base de datos de clientes/ registros	C:\Documents and Settings\All Users\Datos de programa\MySQL\MySQL Server \data C:\Archivos de programa\PostgreSQ\data\base	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (García Martínez, 2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Libreta de direcciones	C:\Documents and Settings\Usuario \Datos de programa\Microsoft\Address Book	
		Correo electrónico	C:\Users\Nombre de usuario \Documents\Archivos de Outlook	
		Registro de bienes/ financieros	C:\Documents and Settings\Nombre de usuario \Mis documentos	
		Log de actividad en Internet	shell:Cache	
		Registros médicos	C:\Documents and Settings\Nombre de usuario \Mis documentos	
	Otros	Libreta de direcciones		
		Biografía		
		Agendas		
		Base de datos de clientes/ registros		
		Correo electrónico		
		Registro de bienes/ financieros		
		Log de actividad en Internet		
		Registros médicos		
		Publicidad en páginas de la web		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz,

		Calendario	/data/data/com.android.providers.calendar/	2007) (García Martínez, 2001) (Solís, 2012)
		SMS&MMS	/data/data/com.android.providers.telephon/	
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	
		Cache	/data/data/com.google.android.location/	
Piratería de software	MAC	Log de chat	Users/Username/Library/Application Support/SecondLife/Avatarname/chat.txt	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
		Historial del navegador/ cache	~/Library/Caches/MS\ Internet\ Cache ~/Library/Caches Safari	
		Archivos y directorios creados por el usuario	~/Documents ~/Images ~/Movies ~/Pictures	
	Windows	Log de chat	C:\Documents and Settings\Usuario\Datos de programa\Pandion	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (García Martínez, 2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Correo electrónico	C:\Users\Nombre de usuario \Documents\Archivos de Outlook	
		Fichero de imagen de certificados de software	C:\Documents and Settings\Nombre de usuario \Mis imágenes	
		Log de actividad en internet	shell:Cache	
	Otros	Log de chat		
		Correo electrónico		
		Fichero de imagen de certificados de software		
		Log de actividad en internet		

		Directorios creados por el usuario y nombres de ficheros de software clasificado con copyright		
Fraude de telecomunicaciones	MAC	Software de clonación	/usr/bin /Applications	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
		Historial del navegador/ caché	~/Library/Caches/MS\ Internet\ Cache ~/Library/Caches Safari	
	Linux	Correo electrónico	~/.thunderbird	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Log de actividad en internet	~/.mozilla	
		Historial del navegador/ cache	~/.mozilla/firefox/CRYPT/Cache	
	Windows	Software de clonación	C:\Documents and Settings\Yanelis\Datos de programa\Seagate DiscWizard C:\Documents and Settings\Yanelis\Datos de programa\Redo Backup & Recovery	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (García Martínez, 2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Correo electrónico	C:\Users\Nombre de usuario \Documents\Archivos de Outlook	
		Log de actividad en internet	shell:Cache	
		Historial del navegador/ cache	Mozilla: offlineCache. Datos para trabajar sin conexión cookies.sqlite. Contiene las cookies formhistory.sqlite. El historial downloads.sqlite. Historial de descargas signons.sqlite. Contraseñas guardadas	
	Otros	Software de clonación		
		Correo electrónico		
		Log de actividad en internet		

		Historial del navegador/ caché		
Abuso/explotación infantil	MAC	Log de chat	Users/Username/Library/Application Support/SecondLife/Avatarname/chat.txt	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Marcas de tiempo en ficheros	–lR /	
		Correo electrónico	~/Library/Mail/*/*/mbox	
		Software de edición y visionado de imágenes	/usr/bin/ /Aplicaciones	
		Log histórico	~/.bash_history	
		Imágenes	~/Pictures	
		Log de actividad en internet	~/Library/Safari/Preferences/Security/Cookies	
		Historial del navegador/cache	~/Library/Caches/MS\Internet\ Cache ~/Library/Caches Safari	
		Videos	~/Movies	
		Archivos y directorios creados por el usuario	~/Documents ~/Images ~/Movies ~/Pictures	
	Linux	Log de chat	~/.purple	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (Mukasey, 2008)
		Marcas de tiempo en ficheros	find/ -atime find/ -mtime find / -ctime	
		Correo electrónico	~/.thunderbird	
		Software de edición y visionado de imágenes	~/.gimp-2.6	
		Log de Historial	~/. Bash_history	
		Imágenes	/home/NombreDeUsuario/imágenes	
		Log de actividad en internet	~/.mozilla	
		Historial del navegador/cache	~/.mozilla/firefox/CRYPT/Cache	
		Videos	/home/NombreDeUsuario/videos	

		Archivos y directorios creados por el usuario	~/ ó /tmp	
	Windows	Log de chat	C:\Documents and Settings\NombreDeUsuario\Datos de programa\Pandion	(ACPO, 1999) (Arquillo Cruz, 2007) (Casey, 2004) (Leigland, 2004) (García Martínez, 2001) (Mukasey, 2008) (Valenzuela Espejo, 2008)
		Marcas de tiempo en ficheros	Ficheros/propiedades/detalles	
		Correo electrónico		
		Software de edición y visionado de imágenes		
		Imágenes	C:\Documents and Settings\NombreDeUsuario \Mis Documentos\Mis imagenes	
		Log de actividad en internet	shell:Cache	
		Log de Historial	Shell:History	
		Historial del navegador/cache	Mozilla: offlineCache. Datos para trabajar sin conexión cookies.sqlite. Contiene las cookies formhistory.sqlite. El historial downloads.sqlite. Historial de descargas signons.sqlite. Contraseñas guardadas	
		Videos	C:\Documents and Settings\ NombreDeUsuario \Mis Documentos\Mis Videos	
		Archivos y directorios creados por el usuario	C:\Documents and Settings\Users\NombreDeUsuario	
	Otros	Log de chat		
		Marcas de tiempo en ficheros		
		Software de cámara digital		
		Correo electrónico		
		Software de edición y visionado de imágenes		
		Imágenes		

		Videos		
		Log de actividad en internet		
		Archivos y directorios creados por el usuario		
		Historial del navegador/cache		
	Android	Contactos	/data/data/com.android.providers.contacts/	(Arquillo Cruz, 2007) (García Martínez, 2001) (Solís, 2012)
		Calendario	/data/data/com.android.providers.calendar/	
		SMS&MMS	/data/data/com.android.providers.telephon/	
		Historial de descargas	/data/data/com.android.providers.downloas/	
		Datos del navegador	/data/data/com.android.providers.browser/	
		Correo electrónico	/data/data/com.google.android.providers.gmail/	
		Cache	/data/data/com.google.android.location/	

Tabla 14. Delito-SO-Componente-Ubicación

Anexo II. Descripción de los escenarios

Nombre del Escenario: Abrir un caso		***Identificador:*** ES2
Objetivo del Escenario: Permitir abrir un caso		
Persona: Investigador		
Iteración: 1ra	***Prioridad:*** 5	***Complejidad:*** Alta
Descripción: El investigador accede al sistema y selecciona la opción Abrir Caso, una vez seleccionada se muestra el listado de casos abiertos y cerrados, con la siguiente información: Fecha de creación Nombre Delito Investigador		

Tabla 15. Escenario "Abrir un caso"

Nombre del Escenario: Guardar un caso		***Identificador:*** ES3
Objetivo del Escenario: Permitir guardar un caso en la base de datos		
Persona: Investigador		
Iteración: 1ra	***Prioridad:*** 5	***Complejidad:*** Alta
Descripción: El investigador selecciona la opción Guardar Caso, una vez seleccionada debe introducir el nombre del caso y seleccionar el estado del caso ya sea abierto o cerrado y guardar.		

Tabla 16. Escenario "Guardar un caso"

Nombre del Escenario: Documentar los datos de la evidencia		***Identificador:*** ES4
Objetivo del Escenario: Permitir registrar los datos de la evidencia		
Persona: Investigador		
Iteración: 2da	***Prioridad:*** 4	***Complejidad:*** Media

Descripción: El investigador selecciona la opción Nuevo Caso, una vez creado el nuevo caso se muestran en la ventana principal los sistemas operativos a analizar, los componentes por cada sistema operativo, su posible ubicación y los datos de la evidencia donde se insertan los elementos encontrados en cada componente analizado y estos son: Herramientas utilizadas Algoritmo utilizado Datos de la evidencia Función resumen Ubicación Evidencia inculpatoria Fecha de la evidencia. Una vez introducidos estos datos se guardan y quedan almacenados en la base de datos.

Tabla 17. Escenario "Documentar los datos de la evidencia"

Nombre del Escenario: Mostrar línea de tiempo		***Identificador:*** ES5
Objetivo del Escenario: Permitir mostrar una línea de tiempo según las fechas de evidencias encontradas		
Persona: Investigador		
Iteración: 2da	***Prioridad:*** 4	***Complejidad:*** Media
***Descripción*:** El investigador selecciona la opción Línea de tiempo, se muestra la opción Crear y Guardar, permitiendo crear la línea de tiempo y guardarla en la ubicación deseada.		

Tabla 18. Escenario "Mostrar línea de tiempo"

Nombre del Escenario: Mostrar reporte del caso		***Identificador:*** ES6
Objetivo del Escenario: Permitir crear el reporte de un caso		
Persona: **Investigador**		
Iteración: 2da	***Prioridad:*** 4	***Complejidad:*** Media
Descripción: El investigador selecciona la opción Reporte del caso. Una vez seleccionado se muestra el contenido del reporte, como: datos de los casos, sistemas operativos implicados, datos de la evidencia; selecciona los datos del caso al que desee crear el reporte. Luego selecciona la opción Crear reporte y lo guarda como archivo de tipo 'pdf' en la ubicación deseada.		

Tabla 19. Escenario "Mostrar reporte del caso"

Nombre del Escenario: Insertar investigador		***Identificador:*** ES7
Objetivo del Escenario: Permitir insertar un investigador en el sistema		
Persona: Administrador		
Iteración: 3ra	***Prioridad:*** 3	***Complejidad:*** Baja
Descripción: El administrador accede al sistema y selecciona la opción Gestión y luego Insertar Investigador, una vez seleccionada la opción Insertar Investigador el sistema mostrará un formulario con los datos a ingresar del investigador: Nombre: Primer Apellido Segundo Apellido C Usuario Contraseña Estado Rol Luego de introducidos los datos se selecciona la opción Aceptar y el sistema guarda los datos del investigador.		

Tabla 20. Escenario "Insertar Investigador

Nombre del Escenario: Modificar investigador		***Identificador:*** ES8
Objetivo del Escenario: Permitir modificar los datos del investigador registrados en el sistema		
Persona: Administrador		
Iteración: 3ra	***Prioridad:*** 3	***Complejidad:*** Baja
Descripción: El administrador selecciona en Gestión de investigador la opción Modificar Investigador; una vez seleccionado se muestra el listado de investigadores que se encuentran en la base de datos, se selecciona la opción **Editar,** la cual permite modificar Nombre, Primer apellido, segundo apellido, CI, usuario, contraseña, estado; cuando se selecciona Aceptar se modifican los datos.		

Tabla 21. Escenario "Modificar investigador"

Nombre del Escenario: Cambiar estado del investigador		***Identificador:*** ES9
Objetivo del Escenario: Permitir cambiar el estado del investigador		
Persona: Administrador		

Iteración: 3ra	***Prioridad:*** 3	***Complejidad:*** Baja
Descripción: El administrador selecciona la opción Cambiar estado del investigador, una vez seleccionada se muestra una lista de investigadores activos y otra de investigadores inactivos, permitiendo pasar de la lista de activos a la de inactivos y viceversa.		

Tabla 22. Escenario "Cambiar estado del investigador"

Nombre del Escenario: Autenticar usuario		***Identificador:*** ES10
Objetivo del Escenario: Permitir el acceso al sistema		
Persona: Investigador-Administrador		
Iteración: 3ra	***Prioridad:*** 3	***Complejidad:***Baja
Descripción: El investigador o administrador introduce su usuario y contraseña para acceder al sistema		

Tabla 23. Escenario "Autenticar usuario"

Anexo III. Diagramas de clases por escenarios

Figura 13. Diagrama de clases "Abrir un Caso"

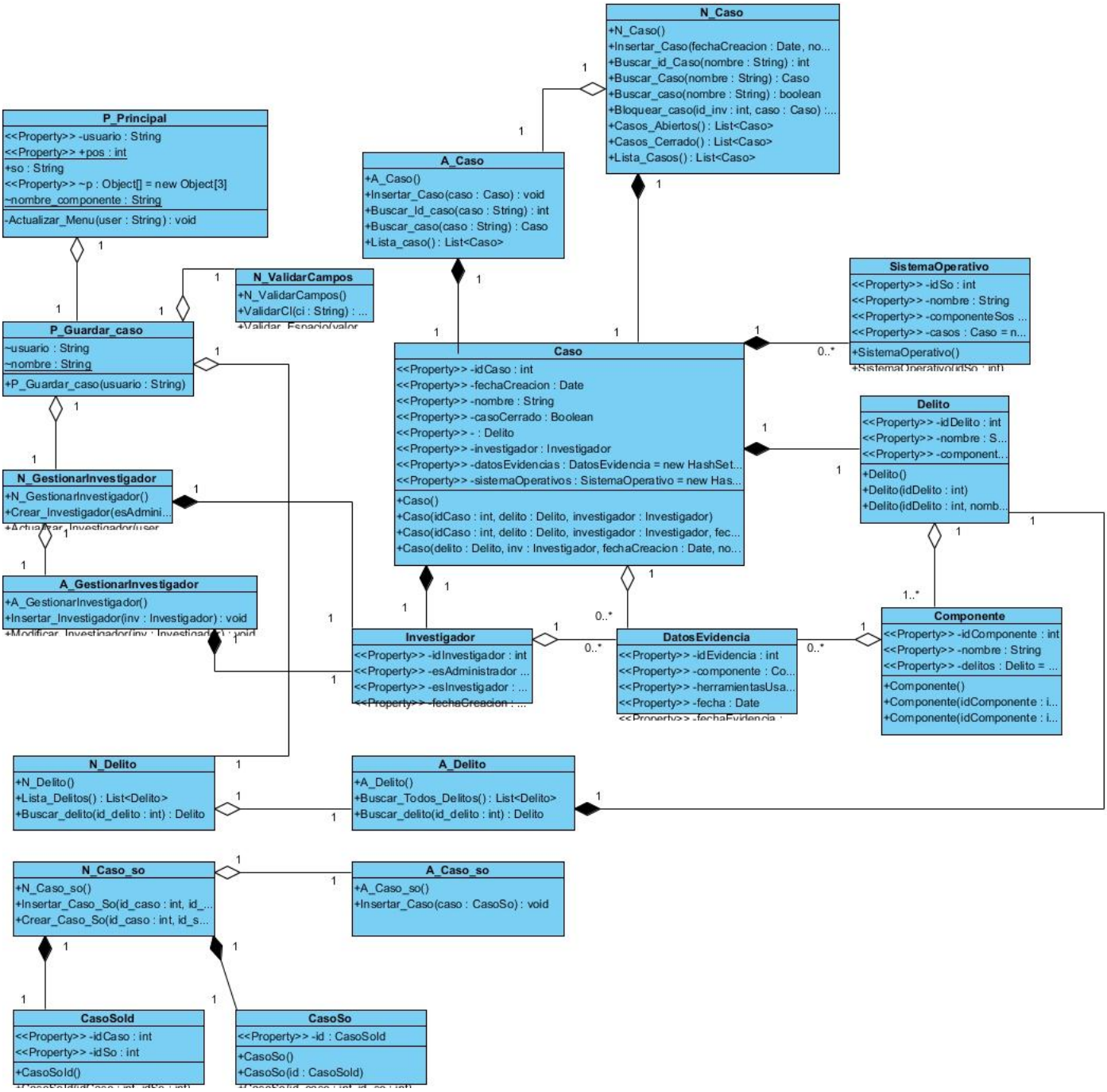


Figura 14. Diagrama de clases "Guardar Caso"

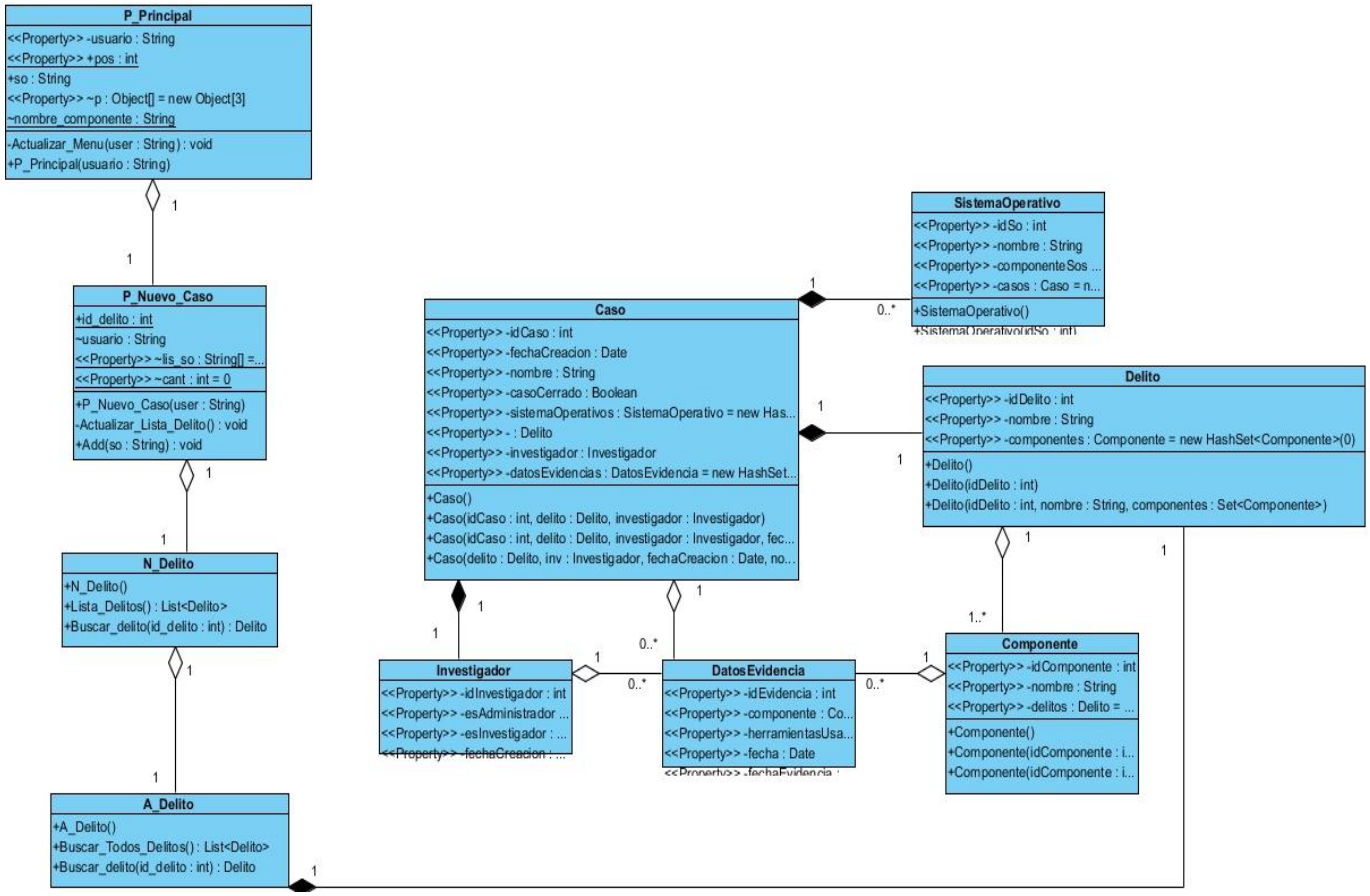


Figura 15. Diagrama de clases "Nuevo Caso"

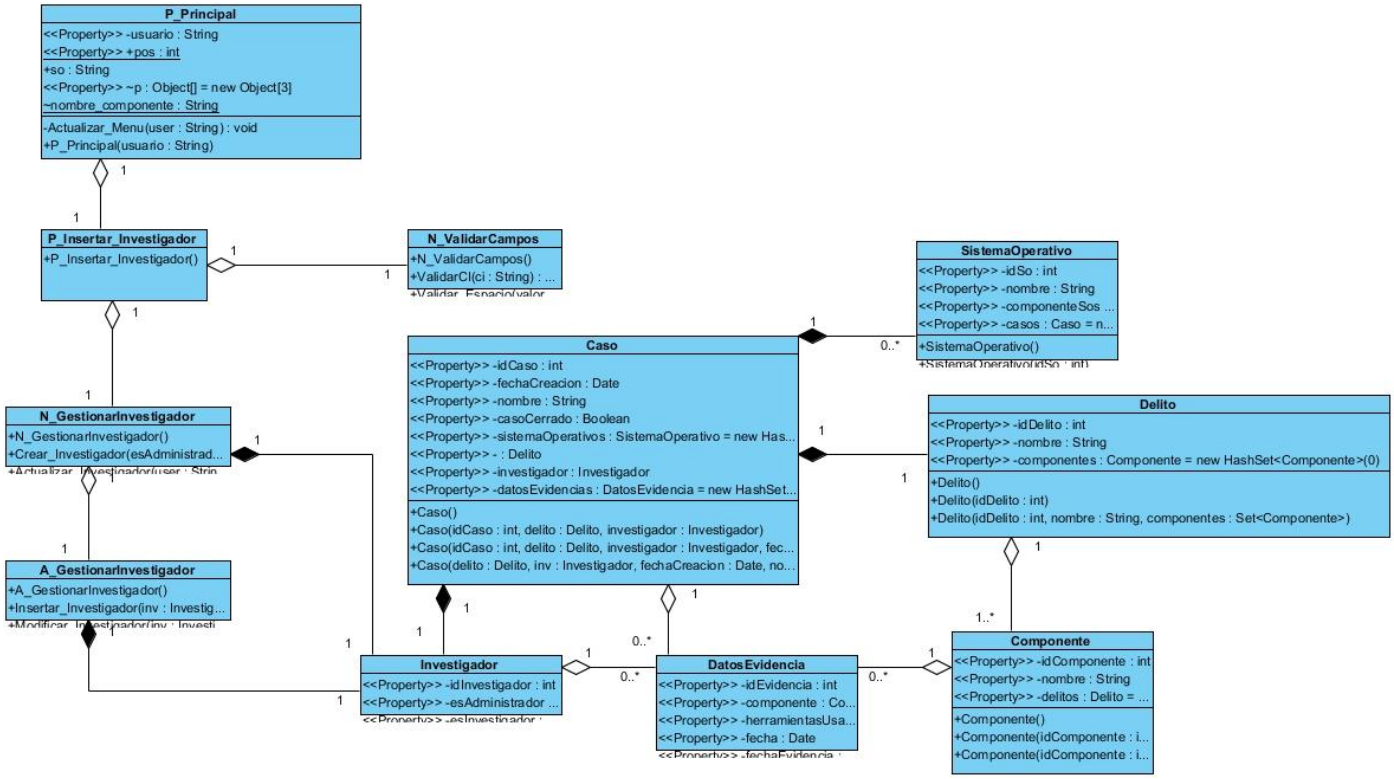


Figura 16. Diagrama de clases "Insertar Investigador"

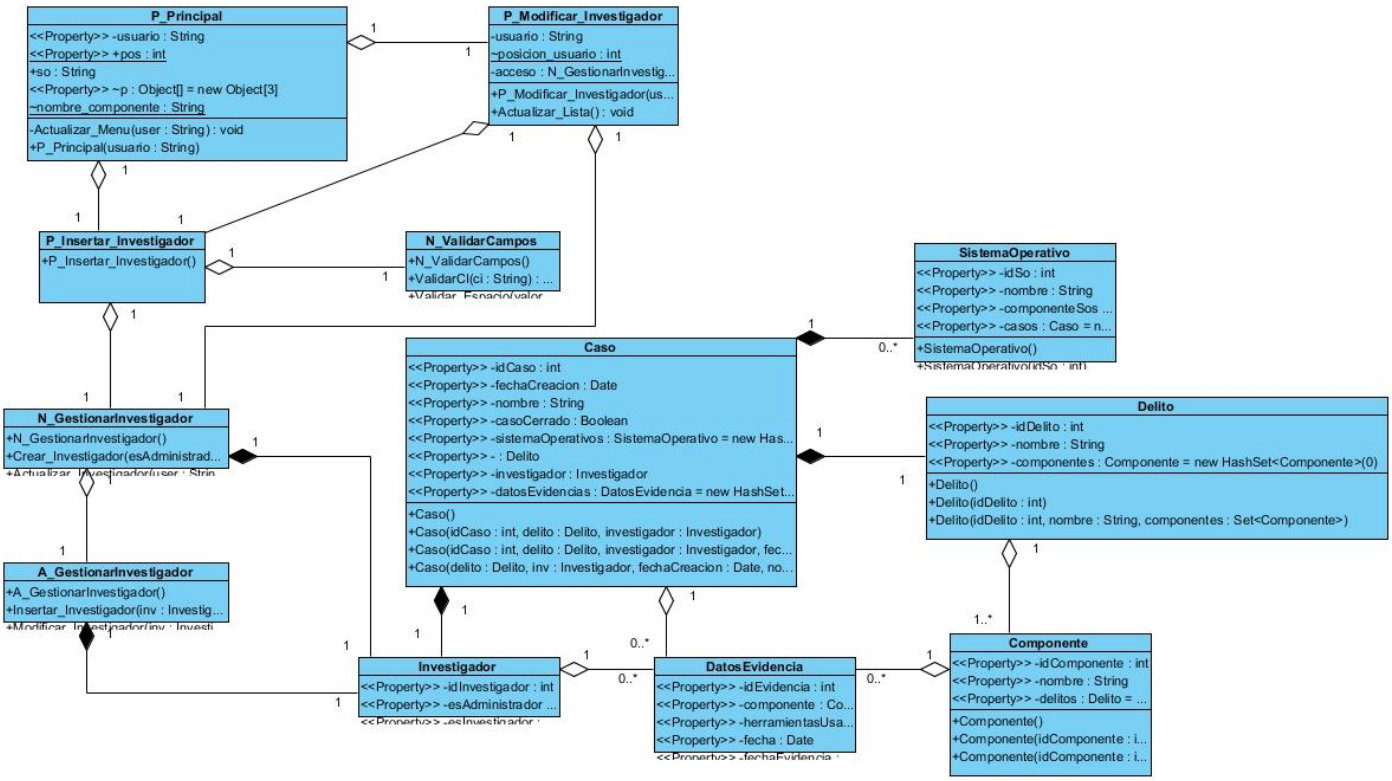


Figura 17. Diagrama de clases "Modificar Investigador"

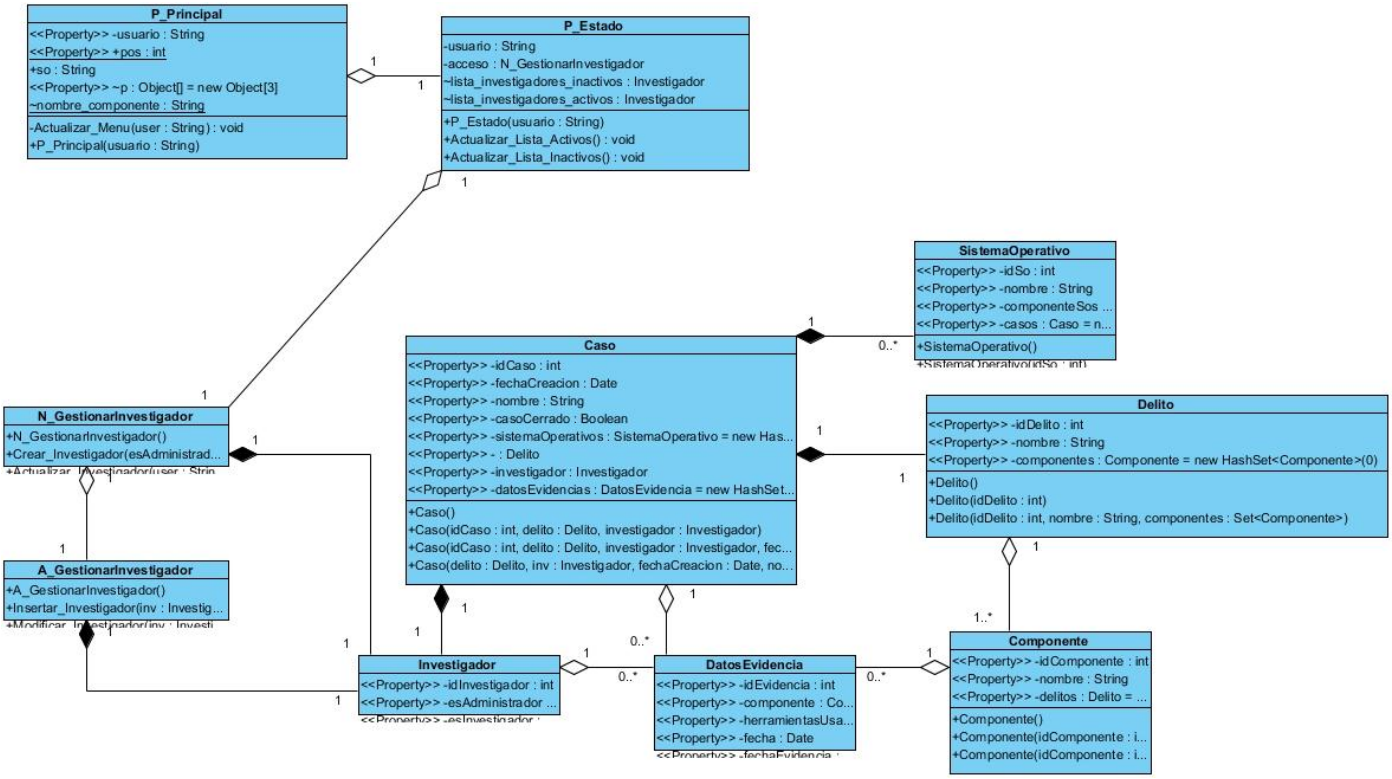


Figura 18. Diagrama de clases "Cambiar Estado del Investigador"

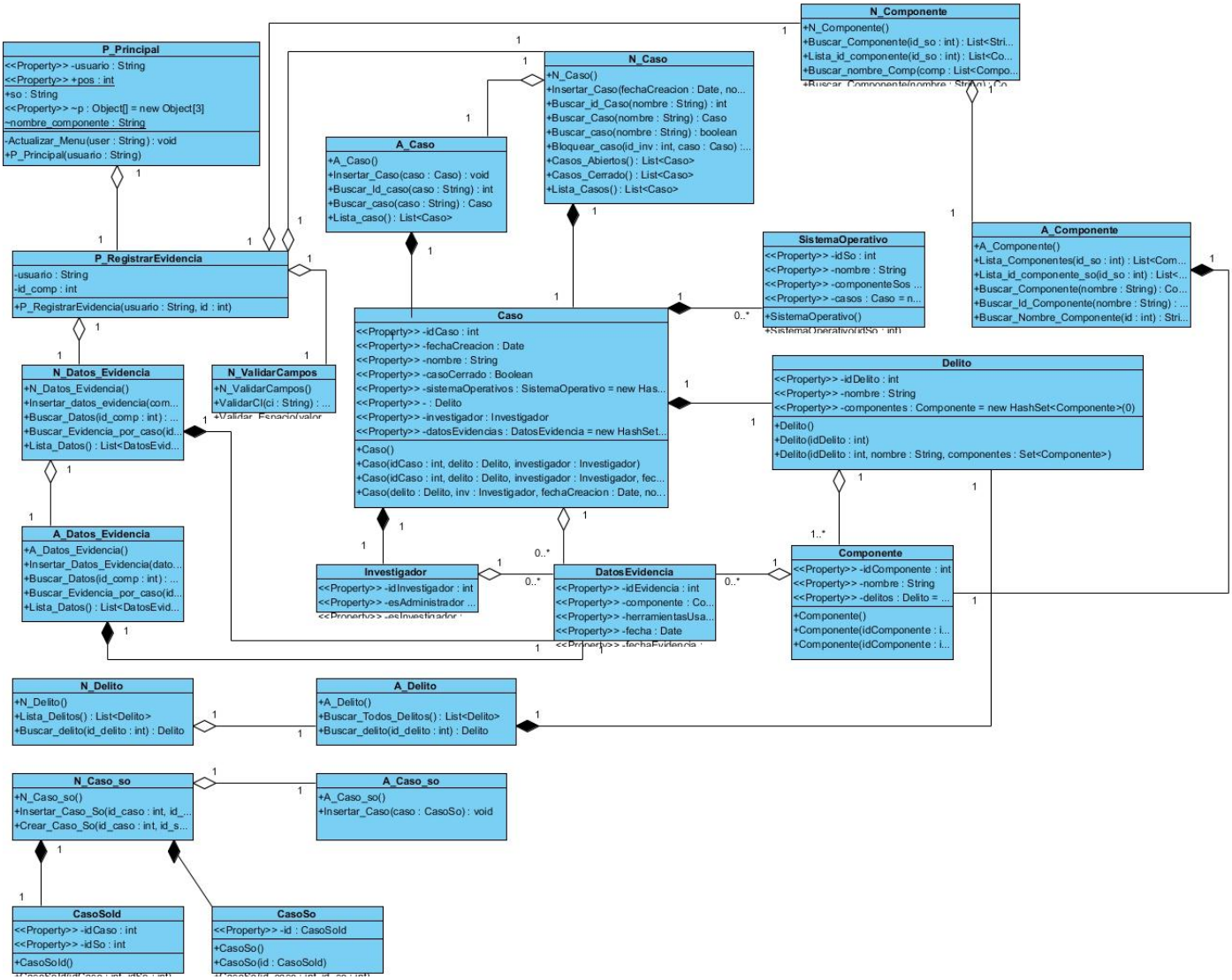


Figura 19. Diagrama de clases "Registrar Evidencia"

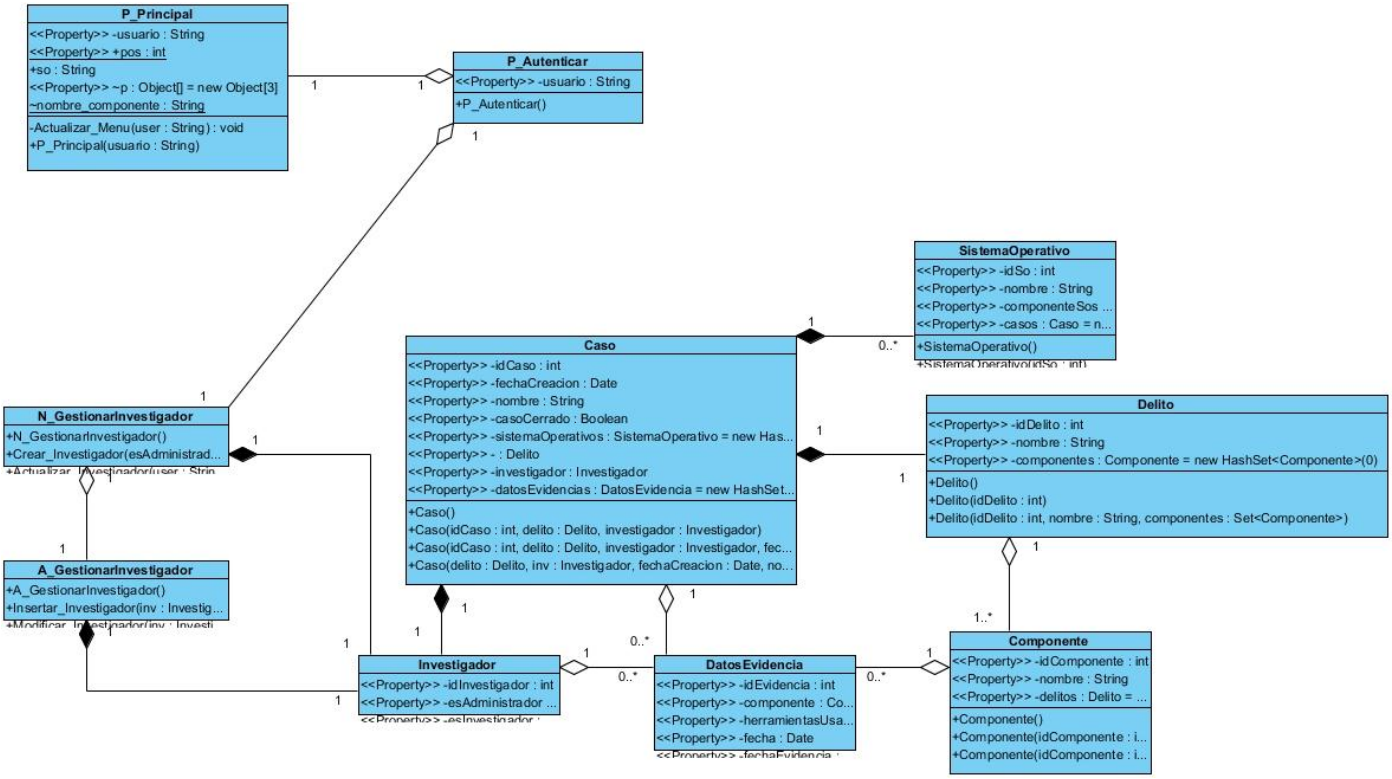


Figura 20. Diagrama de clases "Autenticar Usuario"

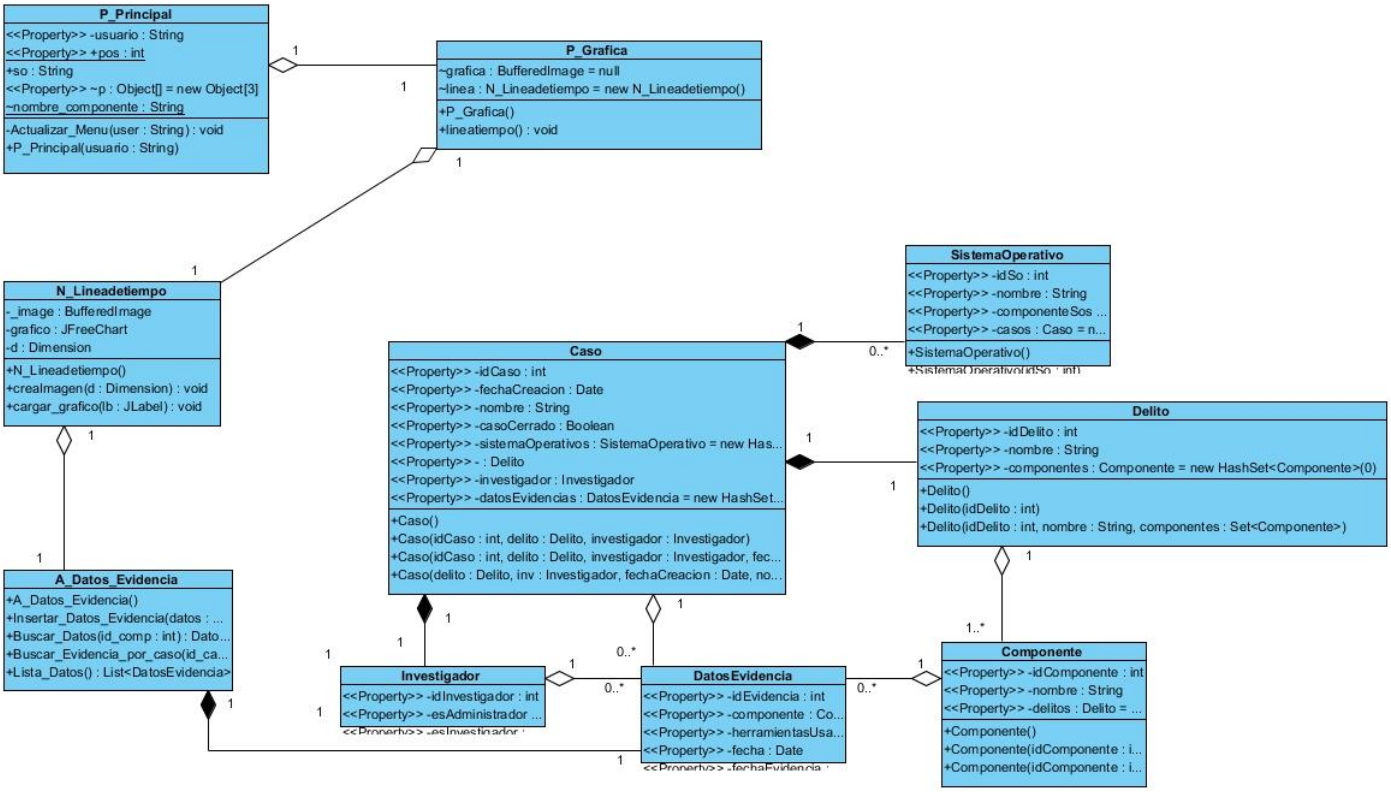


Figura 21. Diagrama de clases " Línea de Tiempo"

Anexo IV. Descripción de los casos de prueba

Escenario	Descripción	Variable 1	Variable 2	Respuesta del sistema	Flujo central
EC 1.1 Autenticarse en el sistema	*Permitir autenticarse con usuario y contraseña*	*V usuario*	*V contraseña*	*Permite autenticarse en el sistema.*	
		I usuario	*V contraseña*	*Muestra un mensaje de error: "El usuario es incorrecto"*	
		V usuario	*I contraseña*	*Muestra un mensaje de error: "La contraseña es incorrecta"*	
		I usuario	*I contraseña*	*Muestra un mensaje de error: "El usuario y contraseña es incorrecto"*	
EC 1.2 Seleccionar opción Aceptar	*Permitir entrar al sistema*			*Permite entrar al sistema, se muestra la pantalla principal*	
EC 1.3 Seleccionar opción Cancelar	*Permitir cancelar la entrada al sistema*			*Se cierra la ventana de autenticación*	

Tabla 24. Caso de prueba "Autenticar usuario"

Escenario	Descripción	Variable 1	Variable 2	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar la opción "Caso"	*Permitir seleccionar opción " Nuevo Caso", "Abrir Caso", "Guardar Caso"*	*NA Caso*		*Muestra un menú desplegable y permite seleccionar la opción " Nuevo Caso", "Abrir Caso", "Guardar Caso"*	*Opción Caso*
EC 1.2 Seleccionar "Nuevo Caso"	*Permitir seleccionar la opción "Nuevo Caso"*	*NA N_Caso*		*Muestra un formulario con un listado de delitos y sistemas operativos*	*Opción Caso/ Nuevo Caso*
EC 1.3 Seleccionar un delito y uno o varios sistemas operativos	*Permitir seleccionar un delito de la lista de delitos y uno o varios sistemas operativos*	*NA L_delitos*	*NA SO*	*Se marca en azul el delito seleccionado y se marca él o los sistemas operativos seleccionados*	*Opción Caso/ Nuevo Caso/ Seleccionar un delito y seleccionar sistema operativo*
		V L_delitos	*I SO*	*Se muestra un mensaje de error "Debe seleccionar uno o varios Sistemas Operativos"*	

		I *L_delitos*	*V* *SO*	*Muestra un mensaje de error: " Debe seleccionar un delito"*	
		I *L_delitos*	*I* *SO*	*Muestra un mensaje de error: " Debe seleccionar un delito y uno o varios sistemas operativos"*	
EC 1.4 Seleccionar la opción "Iniciar"	*Permitir iniciar el caso*	*NA* *Iniciar*		*Muestra en la pantalla principal los delitos seleccionados los componentes asociados al delito, la ubicación, y los datos de evidencia a registrar*	*Opción Iniciar*
EC 1.5 Seleccionar la opción "Cancelar"	*Permitir cancelar la opción nuevo caso*	*NA* *Cancelar*		*Se cancela la opción nuevo caso, y se regresa a la pantalla principal*	*Opción Cancelar*
EC 2.1 Seleccionar opción "Abrir Caso"	*Permitir seleccionar opción "Abrir Caso"*	*NA* *A_Caso*		*Muestra un formulario con la lista de casos abiertos y casos cerrados*	*Opción Caso/ Abrir Caso*
EC 2.2 Seleccionar un caso abierto	*Permitir seleccionar un caso abierto*	*NA* *L_CA*		*Marca en azul el caso seleccionado*	*Opción Caso/ Abrir Caso*
		I *L_CA*		*Muestra un mensaje de error: "Debe seleccionar un caso abierto"*	
EC 2.3 Mostrar casos cerrados	*Permitir mostrar casos cerrados*	*NA* *L_CC*		*Se muestra una lista de casos cerrados*	*Opción Caso/ Abrir Caso*
EC 2.4 Abrir un caso cerrado	*No permitir abrir casos cerrados*	*NA* *L_CC*		*Se muestra un mensaje de error "Debe seleccionar un caso abierto"*	*Opción Caso/ Abrir Caso*
EC 2.5 Seleccionar opción "Abrir"	*Permitir abrir el caso*	*NA* *Abrir*		*Muestra en la pantalla principal los delitos, los componentes asociados al delito, la ubicación, y los datos de evidencia a registrar correspondiente al caso.*	*Opción Caso/ Abrir Caso*
EC 2.6 Seleccionar opción "Cancelar"	*Permitir cancelar la opción abrir caso*	*NA* *Cancelar*		*Se cancela la opción abrir caso y se regresa a la pantalla principal*	*Opción Caso/ Abrir Caso*
EC 3.1 Seleccionar la opción "Guardar Caso"	*Permitir seleccionar opción "Guardar Caso"*	*NA* *G_Caso*		*Muestra un formulario con el campo nombre del caso y estado del investigador*	*Opción Caso/ Guardar caso*
EC 1.3 Introducir nombre	*Permitir escribir el nombre del caso y*	*V* *Nombre_C*	*NA* *Estado*	*Permite guardar el caso*	*Opción Caso/ Guardar caso*

del caso y seleccionar estado	*seleccionar el estado*				
		I *Nombre_C*	*NA* *Estado*	*Se muestra un mensaje de error: "El nombre del caso es incorrecto "*	
		V *Nombre_C*	*I* *Estado*	*Se muestra un mensaje de error: "Debe seleccionar el estado del investigador "*	
		I *Nombre_C*	*I* *Estado*	*Se muestra un mensaje de error: "Los campos son incorrectos"*	
EC1.4 Seleccionar opción "Guardar"	*Permitir guardar un caso*	*NA* *Guardar*		*Permite guardar el caso y se regresa a la pantalla principal*	
EC1.4 Seleccionar opción "Cancelar"	*Permitir cancelar la opción guardar caso*	*NA* *Cancelar*		*Se cancela la opción guardar caso, y se regresa a la pantalla principal*	

Tabla 25. Caso de prueba "Gestionar caso"

Escenario	Descripción	Var1	Var2	Var3	Var4	Var 5	Var 6	Var7	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar "Nuevo Caso"	*Permitir seleccionar la opción "Nuevo Caso"*	*NA* *N_Caso*							*Muestra un formulario con un listado de delitos y sistemas operativos*	*Opción Caso/ Nuevo Caso*
EC 1.2 Seleccionar un delito y uno o varios sistemas operativos	*Permitir seleccionar un delito de la lista de delitos y uno o varios sistemas operativos*	*NA* *L_delitos*	*NA* *SO*						*Se marca en azul el delito seleccionado y se marca él o los sistemas operativos seleccionados*	*Opción Caso/ Nuevo Caso/ Seleccionar un delito y Seleccionar sistema operativo*
		V *L_delitos*	*I* *SO*						*Se muestra un mensaje de error "Debe seleccionar uno o varios Sistemas Operativos"*	
		I *L_delitos*	*V* *SO*						*Muestra un mensaje de error: " Debe seleccionar un delito"*	
		I *L_delitos*	*I* *SO*						*Muestra un mensaje de error: " Debe seleccionar un delito y uno o varios sistemas operativos"*	

EC 1.3 Seleccionar la opción "Iniciar"	*Permitir iniciar el caso*	*NA Iniciar*							*Muestra en la pantalla principal los delitos seleccionados los componentes asociados al delito, la ubicación, y los datos de evidencia a registrar*	*Opción Iniciar*
EC 1.4 Seleccionar la opción insertar datos	*Permitir mostrar los datos de la evidencia a insertar*	*NA Insertar*							*Muestra un formulario con los datos a registrar de la evidencia encontrada*	*Opción Insertar datos*
EC 1.5 Documentar los datos de la evidencia	*Permitir documentar los datos de la evidencia*	*V Herramientas*	*V Algoritmo*	*V Datos de la evidencia*	*V Resumen*	*V Ubicación*	*NA E_inculpatoria*	*V Fecha de la Evidencia*	*Permite documentar los datos de la evidencia*	*Opción Insertar datos*
		I Herramientas	*I Algoritmo*	*I Datos de la evidencia*	*I Resumen*	*I Ubicación*	*I E_inculpatoria*	*I Fecha de la Evidencia*	*Muestra un mensaje de error: "Los datos de la evidencia son incorrectos "*	
		I Herramientas	*V Algoritmo*	*V Datos de la evidencia*	*V Resumen*	*V Ubicación*	*NA E_inculpatoria*	*V Fecha de la Evidencia*	*Muestra un mensaje de error: " El campo herramientas es incorrecto"*	
		V Herramientas	*I Algoritmo*	*V Datos de la evidencia*	*V Resumen*	*V Ubicación*	*NA E_inculpatoria*	*V Fecha de la Evidencia*	*Muestra un mensaje de error: "El campo Algoritmo es incorrecto"*	
		V Herramientas	*V Algoritmo*	*I Datos de la evidencia*	*V Resumen*	*V Ubicación*	*NA E_inculpatoria*	*V Fecha de la Evidencia*	*Muestra un mensaje de error :"El campo Datos la evidencia es incorrecto"*	
		V Herramientas	*V Algoritmo*	*V Datos de la evidencia*	*I Resumen*	*V Ubicación*	*NA E_inculpatoria*	*V Fecha de la Evidencia*	*Muestra un mensaje de error :"El campo resumen es incorrecto"*	
		V Herramientas	*V Algoritmo*	*V Datos de la evidencia*	*V Resumen*	*I Ubicación*	*NA E_inculpatoria*	*V Fecha de la Evidencia*	*Muestra un mensaje de error: "El campo ubicación es incorrecto"*	
		V	*V*	*V*	*V*	*V*	*I*	*V*	*Muestra un*	

		Herramientas	*Algoritmo*	*Datos de la evidencia*	*Resumen*	*Ubicación*	*E_inculpatoria*	*Fecha de la Evidencia*	*mensaje de error: "Debe seleccionar si la evidencia es inculpatoria o no"*	
		V Herramientas	*V Algoritmo*	*V Datos de la evidencia*	*V Resumen*	*V Ubicación*	*NA E_inculpatoria*	*I Fecha de la Evidencia*	*Muestra un mensaje de error: "El campo fecha es incorrecto"*	
EC 1.7 Seleccionar la opción Guardar	*Permitir guardar los datos de la evidencia*	*NA Guardar*							*Permitir guardar los datos de la evidencia*	*Seleccionar opción guardar*
EC 1.8 Seleccionar la opción Cancelar	*Permitir cancelar la opción insertar datos de la evidencia*	*NA Cancelar*							*Se cancela la opción insertar datos de la evidencia*	*Seleccionar opción cancelar*

Tabla 26. Caso de prueba "Documentar los datos de la evidencia"

Escenario	Descripción	Variable 1	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar la opción "Reporte"	*Permitir seleccionar opción "Línea de tiempo", "Reporte del Caso"*	*NA Línea*	*Muestra un menú desplegable y permite seleccionar la opción "Línea de tiempo", "Reporte del Caso"*	*Opción Reporte*
EC 1.2 Seleccionar la opción "Línea de tiempo"	*Permitir seleccionar opción "Línea de tiempo"*	*NA Línea_T*	*Muestra un menú desplegable y permite seleccionar la opción "Línea de tiempo"*	*Opción Reporte/ Línea de tiempo*
EC 1.3 Seleccionar la opción "Crear"	*Permitir crear la línea de tiempo*	*NA C_Línea*	*Permitir crear la línea de tiempo*	*Opción Reporte/ Línea de tiempo/ Crear*
EC 1.4 Seleccionar la opción "Guardar"	*Permitir guardar la línea de tiempo*	*NA G_Línea*	*Permitir guardar la línea de tiempo, muestra la ventana para seleccionar donde se desea guardar*	*Opción Reporte/ Línea de tiempo/ Guardar*

Tabla 27. Caso de prueba "Mostrar línea de tiempo"

Escenario	Descripción	Var 1	Var2	Var3	Var4	Var5	Var6	Var7	Var8	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar la opción "Gestión de investigador "	*Permitir seleccionar opción " Insertar Investigador", Modificar Investigador", "Cambiar estado del investigador "*	*NA Gestionar*								*Muestra un menú desplegable y permite seleccionar la opción " Insertar investigador", "Modificar Investigador", "Cambiar estado del Investigador"*	*Opción Gestión de investigador*
EC 1.2 Seleccionar la	*Permitir seleccionar*	*NA Insertar*									*Opción Gestión de*

opción "Insertar investigador "	*opción " Insertar Investigador"*										*investigador / Insertar investigador*
EC 1.3 Introducir los datos del investigador	*Permitir introducir los datos del investigador*	*V Nombre*	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Permite crear un investigador, muestra un mensaje: "El investigador ha sido creado con éxito"*	*Opción Gestión de investigador / Insertar investigador*
		I Nombre	*I P_Apellido*	*I S_Apellido*	*I CI*	*I Usuario*	*I Contraseña*	*I Estado*	*I Rol*	*Muestra un mensaje de error:"Los datos del investigador son incorrectos "*	
		I Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo nombre es incorrecto"*	
		V Nombre	*I P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo primer apellido es incorrecto"*	
		V Nombre	*V P_Apellido*	*I S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo segundo apellido es incorrecto"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*I CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo carnet de identidad es incorrecto"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*I Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo usuario es incorrecto"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*I Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo contraseña es incorrecto"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*I Estado*	*NA Rol*	*Muestra un mensaje de error :"Debe seleccionar el estado del investigador "*	
		V	*V*	*V*	*V*	*V*	*V*	*NA*	*I*	*Muestra un*	*Opción*

		Nombre	*P_Apellido*	*S_Apellido*	*CI*	*Usuario*	*Contraseña*	*Estado*	*Rol*	*mensaje de error :"Debe seleccionar el rol del investigador"*	*Gestión de investigador / Insertar investigador*
EC 1.4 Seleccionar opción "Aceptar"	*Permitir insertar un investigador*	*NA Insertar*								*Permite insertar un investigador en la base de datos*	*Opción Aceptar*
EC 1.5 Seleccionar opción "Cancelar"	*Permitir cancelar la opción "Insertar Investigador"*	*NA Cancelar*								*Permite cancelar la opción "Insertar Investigador"*	*Opción Cancelar*
EC 2.1 Seleccionar la opción "Modificar investigador "	*Permitir seleccionar opción " Modificar Investigador"*	*NA Modificar*								*Muestra el listado de investigadores que se encuentra en la base de datos*	*Opción Gestión de investigador / Modificar investigador*
EC 2.2 Seleccionar un investigador	*Permitir seleccionar el investigador que se desee modificar*	*NA Seleccionar*								*Marca en azul el investigador seleccionado*	*Opción Gestión de investigador / Modificar investigador*
EC 2.3 Seleccionar la opción cancelar	*Permitir cancelar la opción "Modificar Investigador"*	*NA Cancelar_MI*								*Permite cancelar la opción modificar investigador y regresa a la pantalla principal*	*Opción Gestión de investigador / Modificar investigador /Cancelar*
EC 2.4 seleccionar la opción "Editar"	*Permite editar los datos del investigador*	*NA Editar*								*Muestra el formulario con los datos del investigador a modificar*	*Opción Gestión de investigador / Modificar investigador /Editar*
EC 2.5 Introducir los datos del investigador a editar	*Permitir introducir los datos del investigador a editar*	*V Nombre*	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Permite crear un investigador, muestra un mensaje: "El investigador ha sido creado con éxito"*	*Permite crear un investigador , muestra un mensaje: "El investigador ha sido creado con éxito"*
		I Nombre	*I P_Apellido*	*I S_Apellido*	*I CI*	*I Usuario*	*I Contraseña*	*I Estado*	*I Rol*	*Muestra un mensaje de error:"Los datos del investigador son incorrectos "*	
		I Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo nombre es*	

										incorrecto"	
		V Nombre	*I P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo primer apellido es incorrecto"*	
		V Nombre	*V P_Apellido*	*I S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo segundo apellido es incorrecto"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*I CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo carnet de identidad es incorrecto"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*I Usuario*	*V Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo usuario es incorrecto"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*I Contraseña*	*NA Estado*	*NA Rol*	*Muestra un mensaje de error :"El campo contraseña es incorrecto"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*I Estado*	*NA Rol*	*Muestra un mensaje de error :"Debe seleccionar el estado del investigador"*	
		V Nombre	*V P_Apellido*	*V S_Apellido*	*V CI*	*V Usuario*	*V Contraseña*	*NA Estado*	*I Rol*	*Muestra un mensaje de error :"Debe seleccionar el rol del investigador"*	
EC 2.6 Seleccionar opción "Cancelar"	*Permitir cancelar la opción de introducir los datos del investigador*	*NA Cancelar*								*Permite cancelar la opción de introducir los datos del investigador*	*Opción Cancelar*

EC 3.1 Seleccionar la opción "Cambiar estado del Investigador "	*Permitir seleccionar opción, "Cambiar estado del investigador "*	*NA Gestionar*								*Muestra un listado de los investigadores activos e inactivos*	*Opción Gestión de investigador*
EC 1.1 Seleccionar investigador a "Activar "	*Permitir seleccionar el investigador que se desea activar*	*NA Inactivos*	*NA Activo*							*Pasa del listado de*	*Opción Gestión de investigador*
		I Inactivos	*NA Activo*							*Debe seleccionar un investigador de la lista de inactivos*	*Opción Gestión de investigador*
EC 1.1 Seleccionar investigador a "Desactivar "	*Permitir seleccionar el investigador que se desea Desactivar*	*NA Inactivos*	*NA Activo*							*Pasa del listado de*	*Opción Gestión de investigador*
		I Inactivos	*NA Activo*							*Debe seleccionar un investigador de la lista de inactivos*	*Opción Gestión de investigador*

Tabla 28. Caso de prueba “Gestionar investigador”