Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN TECNOLOXÍAS DA INFORMACIÓN

Desarrollo de una herramienta de código abierto para facilitar la gestión de informes periciales con Python y GTK

Estudante: Mireia Martí Vilas
Dirección: Roberto Rey Expósito

A Coruña, febreiro de 2021.

A mi madre

Agradecimientos

A mis padres, porque sin ellos no habría podido llegar hasta aquí. Y a mi tío Julián, por ofrecerme su ayuda y servirme de inspiración para embarcarme en el maravilloso mundo de la informática.

A mis amigos, especialmente Parafita y Gorka, por ayudarme y apoyarme durante todos estos años.

A mi tutor, por darme la oportunidad de realizar este proyecto y por su tiempo y esfuerzo.

Resumen

La informática forense es una ciencia que forma parte de la seguridad informática y que se centra en identificar, preservar, analizar y presentar las evidencias electrónicas que hayan sido encontradas en un determinado dispositivo. Este proceso tiene que ser documentado minuciosamente por el experto que lo lleva a cabo, y debe adaptarse a normas como UNE 197010:2015 o ISO/IEC 27042:2015. Sin embargo, no existen herramientas informáticas para facilitar esta tarea. Por ello, en este trabajo se desarrolla una herramienta de código abierto para facilitar la elaboración del informe por parte del perito, y la gestión de la documentación relacionada con el caso, al mismo tiempo que mantiene a salvo la información.

Para implementar esta herramienta se ha seguido una metodología de desarrollo incremental, en la que en cada iteración se han ido añadiendo funcionalidades nuevas. Para la implementación se empleó el lenguaje Python 3.0 y la biblioteca GTK+, ofreciendo una herramienta de escritorio potente con una interfaz clara y sencilla.

Abstract

Computer forensics is a science that is part of computer security which focuses on identifying, preserving, analyzing and presenting electronic evidences that have been found on a particular device. This procedure has to be thoroughly documented by an expert who carries it out, and must be adapted to standards such as UNE 197010:2015 or ISO/IEC 27042:2015. However, there are no computer tools to facilitate this task. Therefore, in this work an open source tool is developed to facilitate the preparation of the report by an expert, and the management of the documentation related to the case, while also keeping the information safe.

To implement this tool, an incremental development methodology has been followed, in which new functionalities have been added in each iteration. For the implementation, Python 3.0 language and the GTK+ library were used, offering a powerful desktop tool with a clear and simple interface.

Palabras clave:

- Informática forense
- Informes periciales
- Ciberseguridad
- Código abierto
- Herramienta forense
- Python
- GTK

Keywords:

- Computer forensics
- Forensic reports
- Cybersecurity
- Open source
- Forensics tool
- Python
- GTK

Índice general

Índice de figuras

Índice de tablas

Listings

Capítulo 1

Introducción

En este primer capítulo de la memoria se expone una breve contextualización de este Trabajo Fin de Grado (TFG), detallando los principales motivos que dieron lugar a la creación de la herramienta ForensicReports. También se presentan los objetivos planteados para su implementación, y finalmente la estructura en capítulos de este documento.

1.1 Motivación

La informática forense es una ciencia que forma parte de la seguridad informática y que se centra en identificar, preservar, analizar y presentar las evidencias electrónicas que hayan sido encontradas en un dispositivo [1].

Esta ciencia tuvo que esperar hasta el año 1978 para poder ser empleada a la hora de investigar ciertos delitos. Fue en este año, en el estado de Florida, cuando se reconocen los primeros crímenes cometidos a través de sistemas informáticos. En el año 1983, Peter Norton crea una herramienta para poder recuperar los archivos y aplicaciones eliminados de forma accidental. Esta herramienta es la pionera en la informática forense. A partir del año 1990, el propio FBI comienza a emplear las nuevas tecnologías para localizar evidencias digitales, y fue entonces cuando la informática forense pasó a ser un aliado completamente necesario. En el año 1995, se creó la International Organization of Computer Evidence (IOCE) cuya finalidad consistía en compartir las prácticas y ser un punto de encuentro entre los expertos del mundo. En la actualidad es una de las ciencias que más se solicita para esclarecer hechos y obtener pruebas. A pesar de esto sigue siendo desconocida para el gran público. Sin embargo, está cada vez más presente en nuestras vidas.

El peritaje informático consiste en la recopilación y análisis de datos para dar respuesta a una determinada situación. Es necesario adquirir las evidencias, resguardar la información (a partir de funciones hash y/o cifrado) y analizar los datos obtenidos para obtener las pruebas que soportarán el caso. Todo este proceso ha de ser documentado minuciosamente por el pe-

rito que lo realice, para describir el escenario inicial, dejar constancia de que las evidencias no han sido manipuladas, registrar el procedimiento seguido, y finalmente redactar una conclusión. Dicha documentación se recoge en un informe que, según el caso, será entregado a un tribunal. Por este motivo, es importante que se adapte a unos estándares como la norma UNE 197010:2015 ("Criterios generales para la elaboración de informes y dictámenes periciales sobre Tecnologías de la Información y las Comunicaciones"), y la guía ISO/IEC 27042:2015 que enumera ciertas indicaciones que debe incluir el informe pericial [1]. Sin embargo, la normativa no especifica qué herramienta o herramientas debe utilizar el perito para la redacción del informe, ni para la gestión de la documentación relativa al caso. Típicamente el perito hará uso de distintas herramientas para conseguir este objetivo (procesador de textos, herramientas de cifrado, sistema de archivos del Sistema Operativo (SO), etc). Sin embargo, este proceso es tedioso, en parte repetitivo y propenso a errores. Automatizar este proceso, en la medida de lo posible, sería de gran ayuda al perito al tiempo que aportaría mayor seguridad y más garantías sobre la correcta redacción del informe.

A pesar de la importancia de los informes, y hasta donde abarca el conocimiento de la autora, no existen actualmente aplicaciones específicas para esta tarea. Sí existen muchas herramientas de informática forense, como por ejemplo OsForensics [2], ProDiscover [3], EnCase [4], ForensicExplorer [5], Belkasoft [6] o Cellebrite [7] cuyo objetivo es el análisis de evidencias, la recuperación de datos o el establecimiento de una línea temporal, y que proporcionan entre sus funcionalidades la generación de informes. Sin embargo, esta última no es la funcionalidad principal de estas herramientas. Se trata más bien de informes parciales relacionados con la funcionalidad específica de cada herramienta, y que el perito debe incorporar posteriormente a su informe final. Nótese que el perito típicamente tendrá que hacer uso de varias herramientas y procedimientos para resolver un caso. Por lo tanto, dichas herramientas no se ocupan de la generación del informe final, ni contemplan casos de uso específicos, como pueden ser el cifrado de la información, el control de acceso a la misma, o la exportación del informe a diferentes formatos. Además, la inmensa mayoría de estas herramientas son comerciales, y el precio de sus licencias, muy elevado.

Por lo tanto, en este TFG lo que se pretende es el desarrollo de una herramienta de código abierto, específica para la generación de informes periciales, que solvente las carencias mencionadas y facilite al perito la elaboración del informe, al tiempo que mantiene a salvo la información.

1.2 Objetivos

Este proyecto tiene como objetivo general el desarrollo de una herramienta para la generación de informes periciales a partir de los datos introducidos por el usuario. Los objetivos

específicos del proyecto son los siguientes:

- Facilitar en la medida de lo posible la gestión de la documentación perteneciente a un caso pericial (fotos, datos de las evidencias, etc).
- Proteger el acceso a la documentación mediante contraseña.
- Proporcionar, de forma sencilla para el usuario, un almacenamiento seguro del informe y la documentación relacionada.
- Ofrecer una interfaz sencilla e intuitiva.
- Poder exportar el informe a diferentes formatos una vez se hayan cubierto los campos requeridos.
- Ser distribuido como open source.

1.3 Estructura de la memoria

A continuación se muestra una breve descripción del contenido de cada capítulo en el que se estructura el presente documento:

1. **Introducción**: En este primer capítulo se describe la motivación que dio lugar a la implementación de este proyecto y los objetivos que persigue.
2. **Estado del arte**: En el segundo capítulo se estudian las principales herramientas de informática forense, así como la normativa existente acerca de informes periciales.
3. **Material y métodos**: En el tercer capítulo se exponen los recursos necesarios para llevar a cabo el proyecto y la planificación seguida para su desarrollo.
4. **Desarrollo**: En el cuarto capítulo se especifica la labor realizada en cada una de las iteraciones del proyecto, detallando el análisis, el diseño y la implementación.
5. **Pruebas**: En el quinto capítulo se detallan las pruebas realizadas sobre la herramienta. En concreto, las pruebas unitarias y las pruebas sobre la interfaz.
6. **Conclusiones**: Finalmente, en el sexto capítulo se recogen las conclusiones del proyecto, especificando los resultados obtenidos y las futuras lineas de desarrollo.

Capítulo 2

Estado del arte

PARA el comienzo de este proyecto se ha realizado una búsqueda exhaustiva de información acerca de la informática forense y las principales herramientas que se emplean actualmente. En este capítulo se analizan y comparan dichas herramientas para determinar sus ventajas e inconvenientes. Además, se realiza un estudio de la normativa acerca de informes periciales para determinar la información que gestionará la herramienta a desarrollar.

2.1 Herramientas de informática forense

2.1.1 Belkasoft

Belkasoft [6] es una herramienta comercial multiplataforma. Facilita que un investigador adquiera, busque, analice, almacene y comparta evidencia digital encontrada dentro de ordenadores y dispositivos móviles, memoria RAM y en la nube. Analizará automáticamente la fuente de datos y presentará los artefactos forenses más importantes para que el investigador los revise, examine más de cerca o agregue al informe. Estos informes son ajustables, completos y, lo más importante, válidos para presentar en un tribunal. En el panel principal (ver Figura 2.1) se exponen diversas opciones, entre ellas crear un nuevo caso, o cargar uno existente. En este ejemplo se ha creado un nuevo caso empleando una fuente de datos incluida en la versión de prueba. Una vez adquiridas y analizadas las evidencias, haciendo click derecho sobre los datos, la herramienta da la opción de generar un informe en formato texto, HTML, XML, CSV, EML, PDF, XLSX, DOCX, RTF o KML (ver Figura 2.2)

En la Figura 2.3 se muestra el resultado de generar un informe en HTML de los mensajes de WhatsApp seleccionados en la figura anterior.

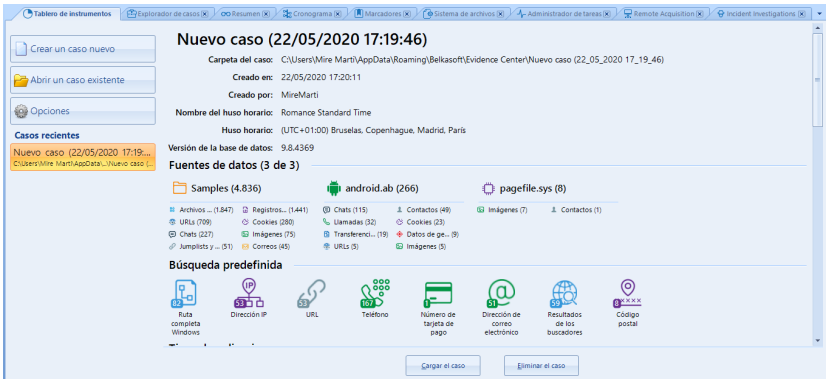


Figura 2.1: Panel principal

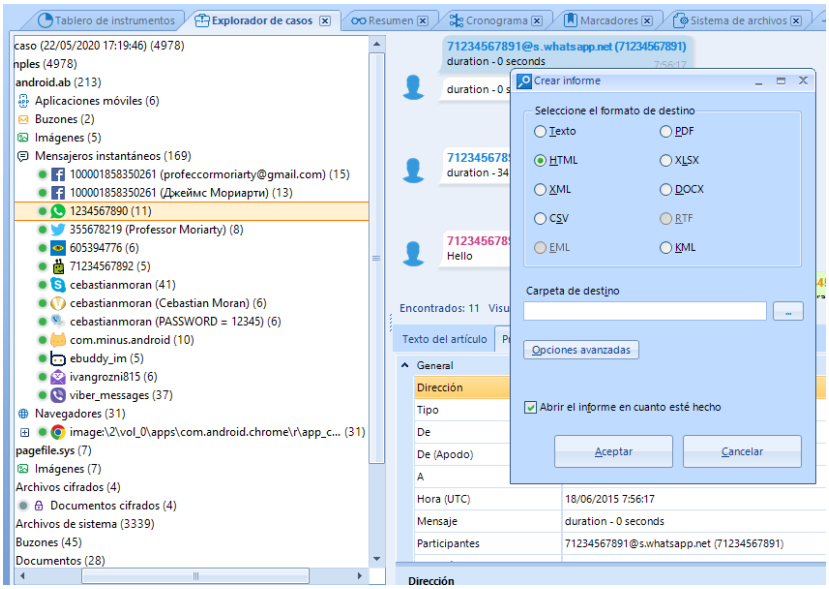


Figura 2.2: Posibles formatos del informe

Finalmente, en las Figuras 2.4, 2.5 y 2.6 se puede ver el mismo informe pero en formato PDF. Como se puede apreciar, el contenido mostrado es el mismo en ambos casos.

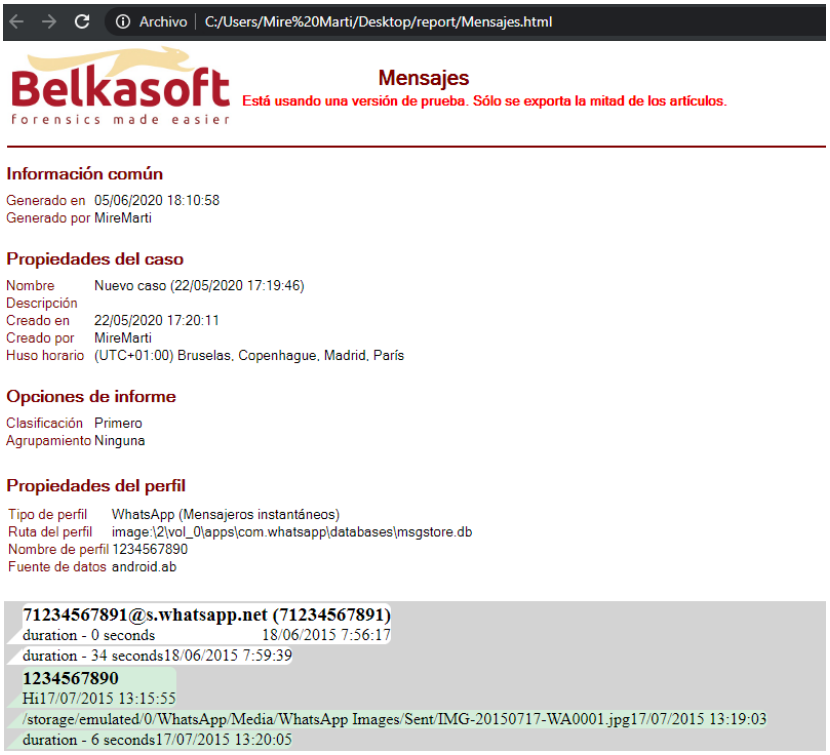
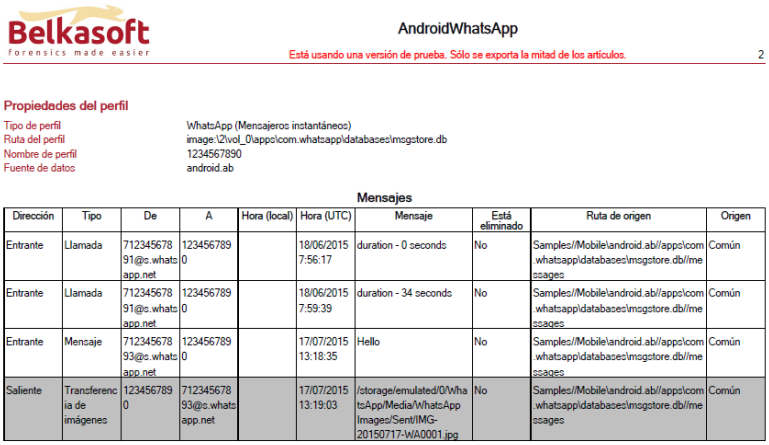


Figura 2.3: Informe en formato HTML

Figura 2.4: Informe en PDF - Parte 1



Belkasoft
forensics made easier

AndroidWhatsApp

Está usando una versión de prueba. Sólo se exporta la mitad de los artículos. 2

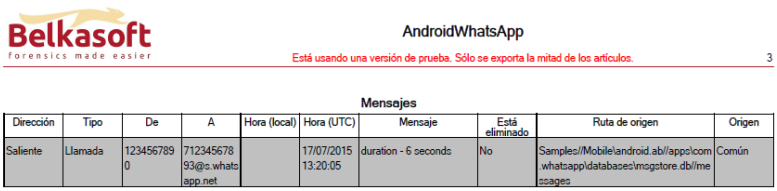
Propiedades del perfil

Tipo de perfil	WhatsApp (Mensajeros instantáneos)
Ruta del perfil	image:\2\vol_0\apps\com.whatsapp\databases\msgstore.db
Nombre de perfil	1234567890
Fuente de datos	android.ab

Mensajes

Dirección	Tipo	De	A	Hora (local)	Hora (UTC)	Mensaje	Está eliminado	Ruta de origen	Origen
Entrante	Llamada	712345678 91@c.whats app.net	123456789 0		18/06/2015 7:56:17	duration - 0 seconds	No	Samples//Mobile\android.ab//apps\com .whatsapp\databases\msgstore.db//me ssages	Común
Entrante	Llamada	712345678 91@c.whats app.net	123456789 0		18/06/2015 7:59:39	duration - 34 seconds	No	Samples//Mobile\android.ab//apps\com .whatsapp\databases\msgstore.db//me ssages	Común
Entrante	Mensaje	712345678 93@c.whats app.net	123456789 0		17/07/2015 13:18:35	Hello	No	Samples//Mobile\android.ab//apps\com .whatsapp\databases\msgstore.db//me ssages	Común
Saliente	Transferenc ia de imágenes	123456789 0	712345678 93@c.whats app.net		17/07/2015 13:19:03	/storage/emulated/0/Wha tsApp/Media/WhatsApp Images/Sent/IMG-20150717-WA0001.jpg	No	Samples//Mobile\android.ab//apps\com .whatsapp\databases\msgstore.db//me ssages	Común

Figura 2.5: Informe en PDF - Parte 2



Belkasoft
forensics made easier

AndroidWhatsApp

Está usando una versión de prueba. Sólo se exporta la mitad de los artículos. 3

Mensajes

Dirección	Tipo	De	A	Hora (local)	Hora (UTC)	Mensaje	Está eliminado	Ruta de origen	Origen
Saliente	Llamada	123456789 0	712345678 93@c.whats app.net		17/07/2015 13:20:05	duration - 6 seconds	No	Samples//Mobile\android.ab//apps\com .whatsapp\databases\msgstore.db//me ssages	Común

Figura 2.6: Informe en PDF - Parte 3

2.1.2 Autopsy

Autopsy [8] es una herramienta forense digital de escritorio basada en Windows que es gratuita, de código abierto y tiene todas las características que normalmente se encuentran en las herramientas forenses comerciales. Es extensible y viene con características que incluyen búsqueda de palabras clave, coincidencia de hash, análisis de registro, análisis web, etc. Esta herramienta tiene una infraestructura de informes extensible que permite crear tipos adicionales de informes para investigaciones. Al final de la investigación, el analista puede exportar el informe a HTML para compartirlo con otras personas, a Excel para copiarlo y pegarlo en otro informe, o incluso puede compartir un subconjunto de los resultados con otro usuario. El módulo de generación de informes se encuentra en la barra de módulos situada en la parte superior de la herramienta, en el botón denominado "Generate Report" (ver Figura 2.7).

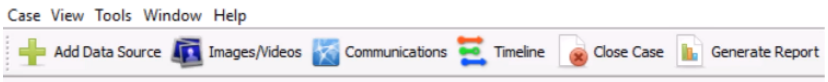


Figura 2.7: Botón para generar informe en Autopsy

Los informes generados en HTML o en Excel especifican los resultados generados durante el análisis. Existen dos modos de obtener los resultados (ver Figura 2.8): generar un informe de todos los elementos hallados, o bien únicamente con los elementos etiquetados. Para el informe HTML se pueden especificar restricciones de uso compartido en el encabezado y pie de página. Además de los dos formatos mencionados, Autopsy también permite generar informes en los siguientes formatos mostrados en la Figura 2.9. Una vez generado el informe en formato HTML, el resultado se representa como se puede apreciar en la Figura 2.10.

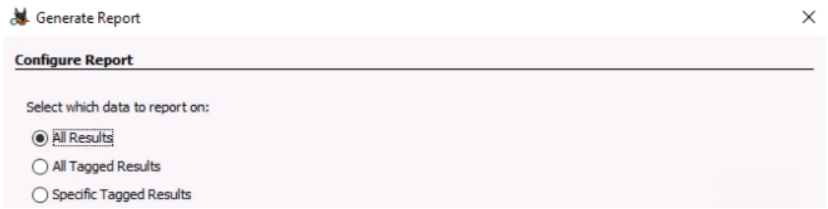


Figura 2.8: Modos de obtener resultados en Autopsy

Los informes de archivo muestran qué archivos hay en el caso. Se representa una línea por cada fichero, y las columnas representan los metadatos (permite seleccionar cuales mostrar, como se aprecia en la Figura 2.11).

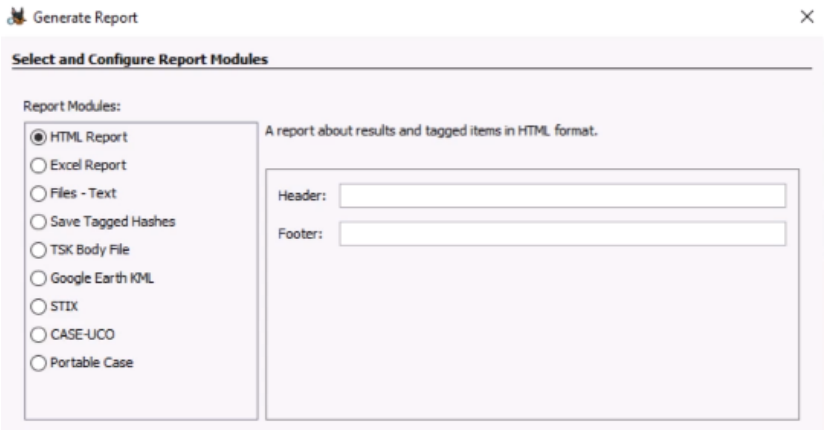


Figura 2.9: Formatos disponibles en Autopsy

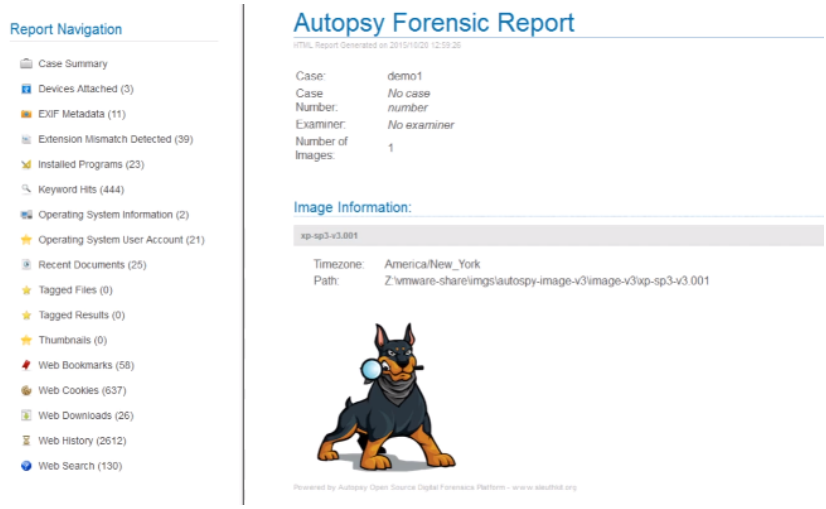


Figura 2.10: Informe en formato HTML - Autopsy

Los informes Keyhole Markup Language (KML) contienen los artefactos Exif, los puntos de seguimiento GPS y las rutas GPS. Además, el módulo de informes permite añadir todos los archivos etiquetados a un conjunto hash, usando la opción "Save Tagged Hashes" (ver Figura 2.12). Una vez generado el informe se obtendrá el enlace para abrirlo.

Finalmente, todos los informes generados se almacenarán en la sección de informes. Un investigador puede generar más de un informe a la vez y editar uno de los existentes, así como crear un nuevo módulo de informes para personalizar el comportamiento para sus necesidades específicas.

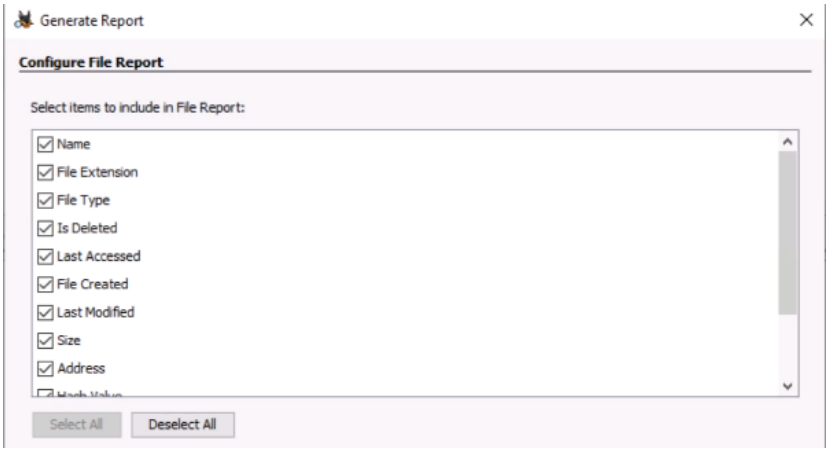


Figura 2.11: Seleccionar datos a mostrar

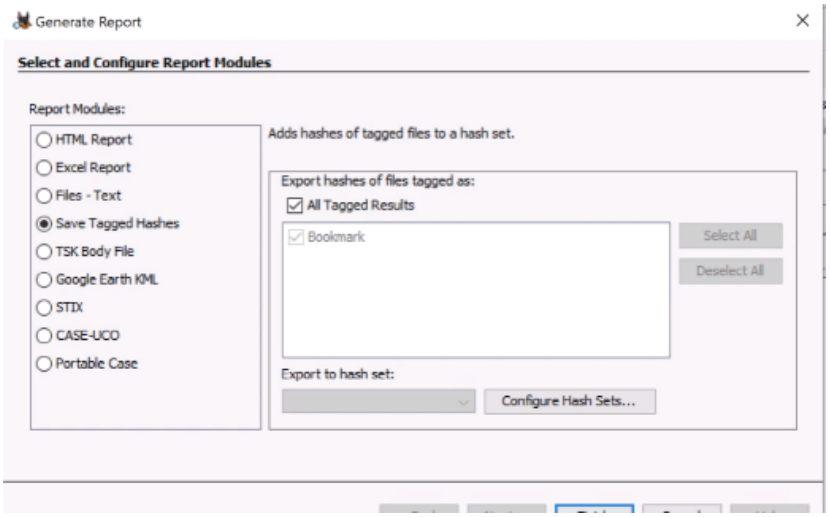


Figura 2.12: Hash Set

2.1.3 OsForensics

OsForensics [2] es una herramienta comercial para Windows que contempla todas las fases de un proceso pericial, desde la creación de un caso y la obtención de todo tipo de evidencias, hasta la generación de un informe final en HTML, que presentará los hallazgos. OsForensics trabaja la información en tres fases:

1. Descubrimiento. La herramienta realiza búsquedas con gran rapidez en toda la superficie del disco o dispositivo elegido, creando además un índice de información. Es capaz

de extraer contraseñas, descifrar archivos y recuperar elementos borrados de diferentes sistemas de archivos: Windows, Mac y Linux.

2. Identificación. Las evidencias y actividades halladas son comparadas mediante su valor hash contra una base de datos. Además, se analizan todos los archivos y permite crear una línea de tiempo de toda la actividad del usuario, para presentarla en orden cronológico.

3. Administración. Finalmente, la suite nos permite organizar todas nuestras evidencias en un guión ordenado, incorporando los datos del examinador forense, presentando los hechos acontecidos y adjuntando datos de otras herramientas forenses si es necesario.

A continuación se mostrará un pequeño ejemplo del funcionamiento de esta herramienta, para estudiar el alcance de la generación de informes. En la Figura 2.13 se puede ver cómo se crea un caso indicando los campos principales. A continuación tienen lugar las fases de descubrimiento e identificación de las evidencias, para finalmente proceder a generar el informe.

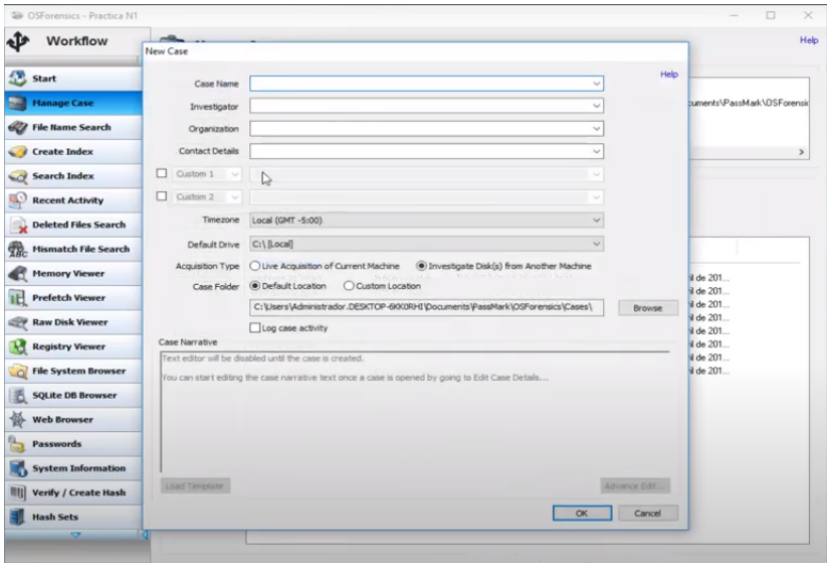


Figura 2.13: Nuevo caso en OsForensics

En la Figura 2.14 se muestran las secciones del análisis que se pueden incluir en el informe y la ruta en la que se va a almacenar. Una vez generado, se almacena en formato HTML.

Finalmente, en la Figura 2.15 se puede ver el informe. Este recopila la información mostrada en la herramienta para poder acceder a ella de forma sencilla. Sin embargo, el formato escogido no permite obtener el informe en un único documento imprimible.

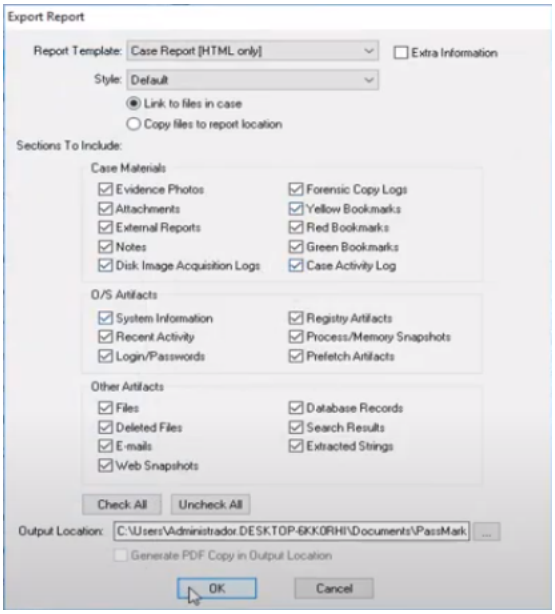


Figura 2.14: Opciones para generar informe en OsForensics

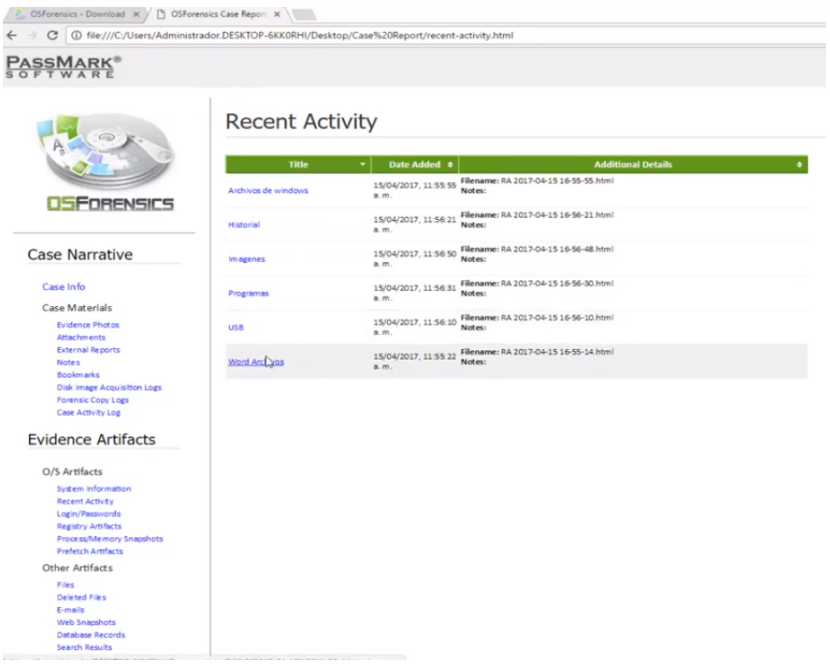


Figura 2.15: Informe en OsForensics

2.1.4 ProDiscover

ProDiscover Forensics [3] es una herramienta de ciberseguridad que permite a los profesionales localizar todos los datos de un disco de almacenamiento informático particular y, al mismo tiempo, proteger la evidencia y crear el informe de documentación utilizado para las órdenes legales. Es multiplataforma y de licencia comercial. Esta herramienta tiene la capacidad de recuperar cualquier archivo eliminado del sistema víctima y examinar el espacio libre. Puede acceder a flujos de datos alternativos de Windows y le permite tener una vista previa y buscar o capturar el proceso (es decir, tomar una captura de pantalla o cualquier otro medio) del Área protegida de hardware (Host Protected Area (HPA)). Con ProDiscover se pueden ir añadiendo comentarios a las evidencias a medida que se investiga un caso (ver Figura 2.16).

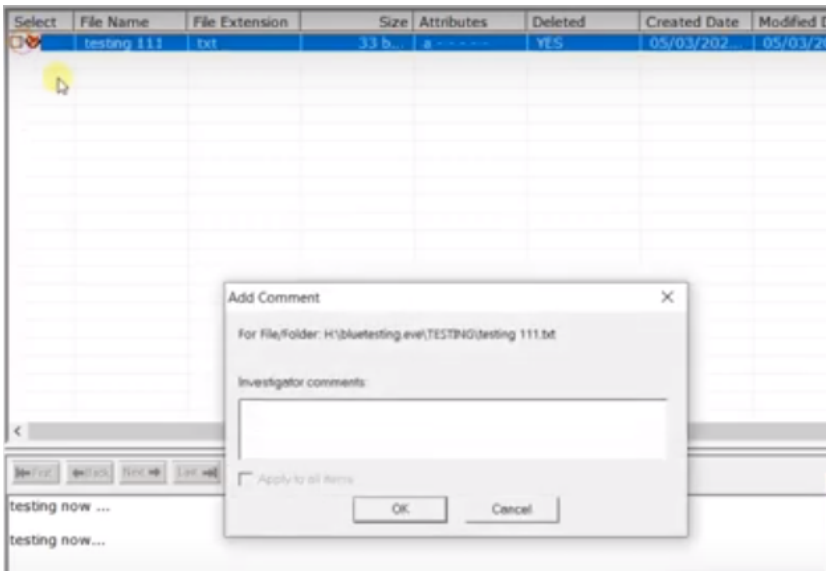


Figura 2.16: Añadir comentario en ProDiscover

El informe puede verse en todo momento desde el apartado “report” de la herramienta. El formato de este informe puede verse en las Figuras 2.17, 2.18 y 2.19. Finalmente, cabe indicar que se puede exportar a formato texto o rtf.

Figura 2.17: Informe en ProDiscover - Parte 1

Figura 2.18: Informe en ProDiscover - Parte 2

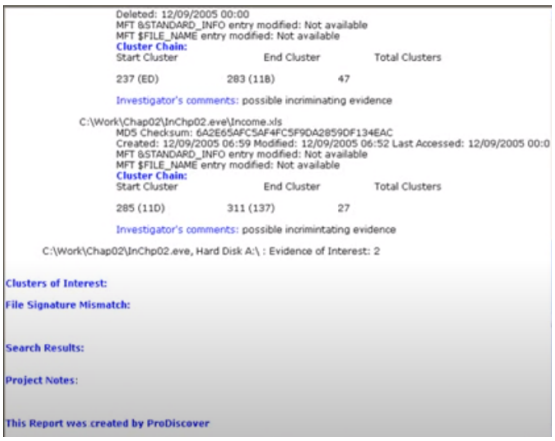


Figura 2.19: Informe en ProDiscover - Parte 3

2.2 Comparación de herramientas

En la Tabla 2.1 se puede ver una comparación entre algunas de las principales herramientas de informática forense.

Nombre	Licencia	SO	Formato/s
OsForensics	Comercial (con versión de prueba)	Windows Vista, Win 7, Win 8, Win 10 Windows Server 2000, 2003, 2008, 2012, 2016, 2019	HTML
Belkasoft	Comercial (con versión de prueba)	Windows (todas las versiones, incluido Windows 10), macOS, sistemas basados en Unix (Linux, FreeBSD, etc.)	TXT, HTML, XML, CSV, PDF, XLSX, DOCX, RTF, KML
ProDiscover	Comercial (con versión de prueba)	Windows, Linux, macOS, Solaris	TXT, RTF
Encase	Comercial	Windows 95/98/NT/2000/XP/ 2003 Server, Linux Kernel 2.4 y superiors, Solaris 8/9 en 32 y 64 bits, AIX, macOS	HTML, RTF
Cellebrite	Comercial	Windows 7, 8, 10	PDF, HTML
Autopsy	GPL	macOS, Solaris, OpenBSD, FreeBSD y Linux	HTML, XLS

Tabla 2.1: Comparación de herramientas

Como se ha mencionado anteriormente, existen muchas herramientas enfocadas a la informática forense. Algunas se centran en la recolección de evidencias, otras en el análisis (de dispositivos, de redes, de imágenes..), clonación de discos, recuperación de datos, etc. En todas las herramientas vistas hasta ahora el objetivo principal no consiste en la generación de informes. Implementan esta funcionalidad para recabar los datos obtenidos durante el análisis, pero estos no se exponen de forma clara y legible.

En el caso de Belkasoft el informe puede exportarse a múltiples formatos, y el sistema operativo puede ser tanto Windows, como macOS o incluso sistemas basados en Unix. Sin embargo, esta herramienta es comercial y su funcionalidad principal no consiste en generar

informes. Por otra parte, Autopsy es de código abierto y multiplataforma, pero tampoco está enfocada en la generación de informes.

Todas las herramientas mencionadas ofrecen informes que exponen los datos recabados durante el análisis, aunque algunas de ellas, como es el caso de Autopsy o Belkasoft, también permiten añadir notas del usuario al informe.

2.3 Estructura del informe pericial

El informe pericial informático deberá estar redactado de forma que pueda ser interpretado correctamente por personas distintas de sus autores. Se requerirá un lenguaje claro, preciso, libre de vaguedades y términos ambiguos, coherente con la terminología empleada en los diferentes capítulos y apartados de los diferentes documentos del informe pericial informático y con una mínima calidad literaria. Los informes generados por las herramientas mencionadas anteriormente no cumplen con las normas siguientes normas:

- UNE 50132:1994 - "Documentación. Numeración de las divisiones y subdivisiones en los documentos escritos". Esta norma hace referencia a la estructura del informe en forma de capítulos y apartados manteniendo una presentación cuidadosa, limpia y ordenada (ver Sección 2.3.1).
- UNE 197010:2015 - "Criterios generales para la elaboración de informes y dictámenes periciales sobre Tecnologías de la Información y las Comunicaciones". Esta norma define el formato del informe pericial (ver Sección 2.3.2).
- UNE 71506:2013 - "Metodología para el análisis forense de las evidencias electrónicas". Define que la fase final del análisis consiste en plasmar toda la información obtenida durante el proceso de análisis de las evidencias en un informe pericial, firmado por el perito informático, dirigido al organismo que solicitó el estudio, manteniendo un equilibrio entre la inteligibilidad y el rigor de lo escrito en el informe (ver Sección 2.3.3).
- Guía ISO/IEC 27042:2015. Enumera ciertas indicaciones que debe incluir el perito informático en su informe pericial, a no ser que existan indicaciones judiciales en contra (ver Sección 2.3.4).

A la vista de las normas anteriores, sería conveniente que un informe pericial se estructurará de acuerdo a los siguientes puntos:

1. Portada
    - Título del informe
    - Organismo o cliente para el que se redacta el informe

- Datos personales del perito
  - Nombre y apellidos
  - Número de identificación del perito
  - Breve curriculum del perito (estudios y experiencia)

2. Índice General

3. Memoria

   - Introducción
     - Incluir identificación del peticionario (juzgado/particular y expediente judicial)
     - Describir objeto principal
     - Detallar los puntos fundamentales
     - Información inicial de la que dispone el perito y su equipo
     - Naturaleza del incidente que va a ser investigado por el perito
     - Fecha, hora y duración del incidente
     - Lugar del incidente
   - Adquisición de evidencias
     - Describir pasos seguidos para la preparación del entorno forense, la adquisición y verificación de las imágenes del equipo afectado
     - Daños en la evidencia digital y sus implicaciones en los siguientes estados del proceso de investigación
     - Cómo se ha mantenido la cadena de custodia
   - Análisis de las evidencias
     - Explicar el procedimiento desarrollado por el cual se ha llegado a las conclusiones finales
     - Hechos sustentados por una evidencia digital y hallados durante la investigación
     - Limitaciones de todos los análisis realizados
     - Detalle de procesos y herramientas utilizadas
     - Interpretación de la evidencia digital por parte del perito
   - Conclusiones
     - Describir de manera detallada los resultados a los que se ha llegado después del análisis

   - Conclusiones y posibles recomendaciones para futuras investigaciones de naturaleza similar

4. Anexos

   - Contenidos técnicos
   - Imágenes
   - Información adicional
   - Glosario de términos

2.3.1 Norma UNE 50132:1994 - Documentación. Numeración de las divisiones y subdivisiones en los documentos escritos

1. Objeto y campo de aplicación

   Esta norma describe un sistema de numeración de las divisiones y subdivisiones en los documentos escritos. Se aplica a todos los documentos escritos tales como libros, trabajos impresos, manuscritos, artículos de revistas y normas.

   La numeración de las diferentes divisiones y subdivisiones de un documento escrito es aconsejable cuando:

   - Aclara la sucesión y la importancia de las diferentes divisiones y subdivisiones, así como sus relaciones.
   - Simplifica la búsqueda y recuperación de determinadas partes del texto, y permite su cita.
   - Facilita las citas o referencias dentro del propio escrito.

2. Numeración de las divisiones y subdivisiones

   (a) La numeración debe realizarse mediante la utilización de cifras arábigas.

   (b) Las divisiones principales (en el primer nivel) de un escrito deben numerarse correlativamente a partir de 1.

   (c) Cada división principal puede subdividirse (en el segundo nivel) en un número cualquiera de subdivisiones numeradas correlativamente a partir de 1. Esta forma de división y numeración puede continuar hasta cualquier nivel (tercer nivel o sucesivos). No obstante, es conveniente limitar el número de niveles a fin de que los números de las distintas partes sean fáciles de identificar, leer o citar.

   (d) La separación de las diversas subdivisiones que forman parte de una misma división principal, se realiza intercalando un punto entre las cifras representativas

de las mismas. No debe usarse el punto a continuación del número que designa el último nivel.

(e) Puede atribuirse la cifra 0 a la primera división de cada nivel, cuando constituya una introducción, un prefacio, un prólogo, un preámbulo o cualquier otra parte de tipo similar.

3. Citación de los números de las divisiones y de las subdivisiones dentro del propio texto
   Los números de las divisiones o de las subdivisiones se citan dentro del propio texto, en la forma que se indica en los siguientes ejemplos:

   - … en el 4 …
   - … véase 9.2 …

4. Enunciación
   Para la enunciación de un número de división o de subdivisión no se tienen en cuenta los puntos. Ejemplos:

   - 2 “dos”
   - 2.1.1 “dos uno uno”

Para más información sobre esta norma consultar [9].

2.3.2 Norma UNE 197010:2015 - Criterios generales para la elaboración de informes y dictámenes periciales sobre Tecnologías de la Información y las Comunicaciones

En su introducción, la norma enumera los principios que deben respetarse durante la selección, obtención, presentación y almacenamiento de evidencias, físicas o digitales:

- Relevancia: es la propiedad que resalta unas evidencias sobre otras en función de su trascendencia y el valor que aportan en el informe pericial.
- Fiabilidad: capacidad de que se puedan reproducir los resultados del proceso de forma consistente por investigadores independientes, a partir de las mismas evidencias.
- Suficiencia: que la evidencia presentada sea representativa, acorde y proporcionada al objeto de lo que se quiere demostrar.
- Oportunidad: que sea representativa de las circunstancias y momento temporal en que se presenta como prueba, y pueda ser trascendente en el juicio.

En su apartado 4, la norma describe los requisitos generales del informe pericial en materias tecnológicas, apoyándose en la norma UNE 197001:2011. Resalta en este aspecto la inclusión del sello de visado del colegio profesional correspondiente, cuando proceda. Se hace también mención a que la firma del dictamen por el perito, si se proporciona el mismo en soporte digital, debe ir incorporada digitalmente. Por último, la norma enumera las evidencias digitales mínimas que deben contener los informes o dictámenes periciales TIC, según su tipología:

- Sistemas de información:
  - Descripción del sistema de información analizado.
  - Gestión de la cadena de custodia.
  - Fecha y hora de intervención.
  - Condiciones de funcionamiento del sistema.
  - Medidas que se han tomado para salvaguardar el sistema de información.
  - Procedimiento y documentación.
  - Política de seguridad de la instalación donde está operando el equipo, incluyendo copias de seguridad.
  - Identificación del personal con acceso al equipo, como mínimo el administrador el sistema.
  - Topología de red, cortafuegos, NAT (Network Address Translation), VPN (Virtual Private Network), enlaces a internet, entre otros.
  - Normativa aplicada en la instalación afectada.
- Autenticación del correo electrónico:
  - Valorar la seguridad del mecanismo de firma electrónica del correo.
  - Si no va firmado, hacer análisis de la cabecera o ver si existe un tercero con copia del mensaje.
  - Cotejo de las cabeceras del correo electrónico con los históricos de los servidores utilizados.
  - Informe del proveedor de internet, si procediera.
- Delitos contra la propiedad intelectual e industrial en formato digital. Identificación, manipulación o utilización de:
  - Componentes hardware.
  - Elementos software.

  - Documentos digitales, películas, vídeos, música y juegos.
  - Patentes y propiedad intelectual relacionadas con las TIC.

- Utilización e identificación de metadatos encontrados en:
  - Correos electrónicos.
  - Fotografías y documentos gráficos.
  - Documentos electrónicos de texto.

- Contenido web :
  - Captura de la pantalla en modo gráfico.
  - Acta testimonial del contenido.
  - acceso a la página web en cuestión.

- Soporte de almacenamiento digital (discos duros, pendrives, memorias SD, etc.) o inventario del contenido:
  - si se ha iniciado o continuado la cadena de custodia.
  - si se ha realizado copia forense del componente original.
  - si se ha aplicado las claves HASH11 al elemento original y a la copia.

Para más información sobre esta norma consultar [10].

2.3.3 Norma UNE 71506:2013 - Metodología para el análisis forense de las evidencias electrónicas

Esta norma tiene como objetivo definir el proceso de análisis forense dentro del ciclo de gestión de evidencias electrónicas. En ella se establecerá una metodología para la preservación, adquisición, documentación, análisis y presentación de las evidencias electrónicas.

Se incluyen una serie de anexos, entre ellos un modelo de informe pericial, en el que se incluyen los siguientes apartados:

1. Asunto
2. Evidencias/muestras recibidas
3. Resolución o estudios efectuados sobre las evidencias/muestras
4. Situación final de las evidencias/muestras
5. Conclusiones finales
6. Anexos del informe

El segundo anexo trata genéricamente las competencias con las que ha de contar el personal involucrado en las diversas fases del análisis forense, separadas en diferentes categorías: competencias técnicas, profesionales y personales.

El último anexo descrito en la norma se refiere al equipamiento para el análisis forense de las evidencias electrónicas. Se debe contar con herramientas tanto hardware como software reconocidas por la comunidad forense internacional, aún no existiendo una normalización.

Esta norma detalla cada fase del análisis forense. En concreto, en el apartado de documentación se puede encontrar la siguiente información: cualquier análisis forense requerirá un control sobre las evidencias que van a ser sometidas a estudio. Por tanto, se documentará todo el procedimiento desde que se inicia el análisis hasta que acaba a través de la redacción del informe pericial a enviar al solicitante, indicando todos los procesos y herramientas utilizadas y el momento en el que fueron ejecutados dichos procesos, siguiendo una secuencia temporal definida con vistas a elaborar un registro auditable.

Consecuentemente, la cadena de custodia debe tener implementado un sistema de gestión documental donde van a quedar reflejados todos los pasos llevados a cabo. Esta gestión documental incluirá, entre otros, los siguientes documentos: documento de recepción de evidencias electrónicas, registro de la documentación recibida, registro de las evidencias, registro del tratamiento inicial en el que se describirá el proceso de clonado, registro de situación de evidencias, registro de tareas del análisis inicial, y registro de tareas del análisis de datos definitivo.

Para más información sobre esta norma consultar [11].

2.3.4 ISO/IEC 27042 - Guía para el análisis y la interpretación de las evidencias digitales

Esta norma proporciona una guía para el análisis e interpretación de la evidencia digital. Fue publicada en junio de 2015 (en inglés). Provee información sobre cómo adelantar un análisis e interpretación de la evidencia digital potencial en un incidente con el objeto de identificar y evaluar aquella que se puede utilizar para ayudar a su comprensión. Ofrece un marco común para el análisis e interpretación de la gestión de incidentes de seguridad, que pueda utilizarse para implementar nuevos métodos. También introduce una serie de definiciones relevantes para la práctica del análisis forense digital.

Adicionalmente, la norma trata los modelos analíticos que pueden ser usados por los peritos informáticos forenses en sistemas estáticos o activos y las consideraciones, a tener en cuenta en cada caso, en especial atención a incidentes en sistemas vivos o activos como: dispositivos móviles, sistemas cifrados, redes, etc. Se definen dos formas de adelantar el análisis en vivo: Sistemas que no pueden ser copiados o que no permiten crear una imagen: existe el riesgo de perder la evidencia digital cuando se está copiando. Es importante ser cuidadoso para minimizar el riesgo de daño de la evidencia y asegurar que se tiene un registro completo de los procesos. Sistemas que permiten copiar o realizar una imagen: examinar el sistema interactuando u observándolo en su operación. Ser cuidadoso para emular el hardware o software del entorno original, usando máquinas virtuales verificadas, copias del hardware original con el fin de permitir un análisis lo más cercano posible al real. Por otro lado, se detalla el contenido de los resultados del análisis en el informe pericial y sus consideraciones legales. Finalmente, recoge las competencias de los peritos forenses: formación, aprendizaje, habilidades, objetividad y ética profesional.

En cuanto al contenido del informe pericial, se enumeran ciertas indicaciones que debe incluir el perito informático en su informe pericial, a no ser que existan indicaciones judiciales en contra. Dichas indicaciones son las siguientes:

- Calificaciones del perito informático.
- Información inicial de la que dispone el perito informático y su equipo.
- Naturaleza del incidente que va a ser investigado por el perito informático.
- Fecha, hora y duración del incidente.
- Lugar del incidente.
- Objetivos de la investigación.
- Miembros del equipo de la investigación supervisados por el perito informático.

- Fecha, hora y duración de la investigación.
- Lugar de la investigación.
- Hechos sustentados por una evidencia digital y hallados durante la investigación.
- Daños en la evidencia digital y sus implicaciones en los siguientes estadios del proceso de investigación.
- Limitaciones de todos los análisis realizados.
- Detalle de procesos y herramientas utilizadas.
- Interpretación de la evidencia digital por parte del perito informático.
- Conclusiones y posibles recomendaciones para futuras investigaciones de naturaleza similar.

Para más información acerca de esta guía consultar [1].

Capítulo 3

Material y métodos

En este capítulo se especifican los recursos empleados para llevar a cabo el proyecto, teniendo en cuenta el material, los métodos y la planificación seguida.

3.1 Material

El material necesario para realizar este proyecto es el siguiente:

- Recursos hardware: ordenador (sistema operativo indiferente, puesto que la herramienta se desarrollará en un lenguaje multiplataforma).
- Recursos software: Python 2 (2.6 o posterior) o Python 3 (3.1 o posterior) y GTK+3.

Para la realización de los prototipos se ha empleado Balsamiq, una herramienta de licencia comercial con versión de prueba, pensada para facilitar el diseño de la interfaz de usuario y permitir al desarrollador tener una visión más clara y estructurada de la aplicación. Los wireframes pueden exportarse como PNG o PDF. Una de las funcionalidades clave es la vinculación interactiva de prototipos, la cual permite enlazar unas pantallas con otras y ofrecer una demostración del funcionamiento mucho más clara y precisa.

Para realizar pruebas sobre herramientas de informática forense existentes se ha empleado la versión de prueba de Belkasoft Evidence Center 2020 y Autopsy 4.15.0

Finalmente, para almacenar el código en un repositorio y mantener a salvo la información, se ha empleado el gestor de repositorios GitLab.

3.2 Métodos

Por otra parte, en cuanto al método empleado se seguirá una metodología de desarrollo iterativo e incremental (ver Figura 3.1). Esta metodología consiste en planificar el proyecto en bloques temporales denominados iteraciones. En cada iteración se repite un proceso de trabajo

similar (análisis, diseño, codificación y pruebas) para proporcionar un resultado completo sobre el software final. De esta manera, es posible comenzar con las funcionalidades más básicas del sistema, para posteriormente ir añadiendo nuevas funcionalidades más complejas.

Las ventajas que esta metodología ofrece son, entre otras, la posibilidad de tomar decisiones tras cada iteración, puesto que una vez finalizada se concierta una reunión con el cliente (en este caso el tutor) y se revisa el producto obtenido. Además, permite gestionar la complejidad del proyecto ya que en cada iteración sólo se trabaja en los requisitos que aportan más valor en ese momento.

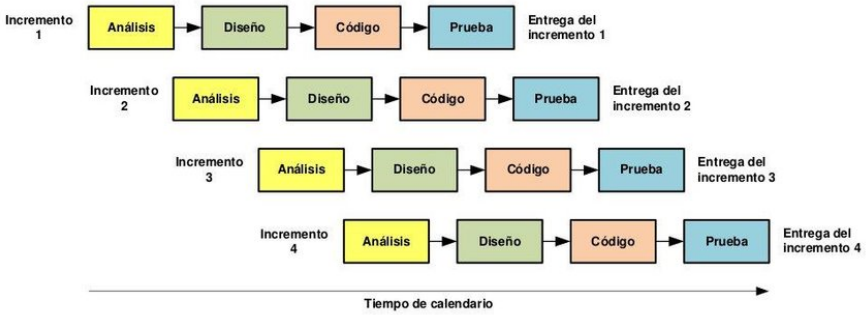


Figura 3.1: Esquema de desarrollo iterativo e incremental

3.3 Planificación

Las fases principales del trabajo se enumeran a continuación.

1. Búsqueda de información: se realizará una búsqueda exhaustiva acerca de la informática forense, las herramientas existentes, la tecnología y metodología más adecuadas al proyecto, etc...

2. Elaboración del estado de la cuestión: este apartado consistirá en recopilar la información necesaria para conocer el estado actual del campo de estudio tratado. En este caso, herramientas relacionadas con la informática forense y la normativa aconsejada para redactar informes periciales.

3. Desarrollo de la herramienta

    - Análisis de los casos de uso: se determinarán y definirán las funcionalidades principales de la herramienta.
    - Definición de las iteraciones: se organizarán los casos de uso obtenidos en la fase anterior para definir todas las iteraciones que se llevarán a cabo durante el desarrollo.

- Prototipo y diseño: se crearán los mockups y se estudiarán las decisiones de diseño correspondientes a los casos de uso asignados a cada iteración.
- Implementación: se implementará el código necesario para obtener las funcionalidades esperadas en cada iteración.
- Pruebas: se realizarán pruebas sobre las funcionalidades implementadas en cada iteración.

4. Documentación

- Elaboración de la memoria: se redactará la memoria del proyecto a partir de todo el trabajo realizado.
- Elaboración de un manual de usuario: se redactará un breve manual de usuario especificando cómo utilizar la herramienta.

Como se ha mencionado en la Sección 3.2, la metodología seguida planifica el proyecto en iteraciones. Por lo tanto, las fases "Prototipo y diseño", "Implementación" y "Pruebas" se repetirán para cada una de las iteraciones definidas durante la fase "Definición de las iteraciones".

La supervisión del proyecto se realiza mediante revisiones periódicas al final de cada iteración de desarrollo. Durante las mismas, el director del proyecto se reúne con el equipo de desarrollo (la estudiante) para evaluar el resultado de la iteración y definir futuros incrementos de funcionalidad. En este proyecto, el papel de director de proyecto se compartió entre el tutor del proyecto y la estudiante. Debido a circunstancias externas, la mayoría de las reuniones se realizaron por videoconferencia.

En la Figura 3.2 se muestra el Diagrama de Gantt correspondiente a la planificación realizada. Cabe destacar que la tarea denominada como "Documentación - Memoria" se realiza de forma simultánea al resto de tareas, dado que se considera conveniente ir completando la memoria a medida que se avanza en el desarrollo.

Por motivos externos al proyecto, la planificación se vió afectada incrementando la duración del desarrollo. Como se puede ver en la Figura 3.3, la duración de la iteración 5 se incrementó en 10 días, y además, se implementó una nueva iteración. Finalmente se añadió un período de correcciones para perfeccionar detalles de la aplicación. Cabe mencionar que durante los meses de agosto y septiembre el proyecto se paralizó por vacaciones del equipo de desarrollo.

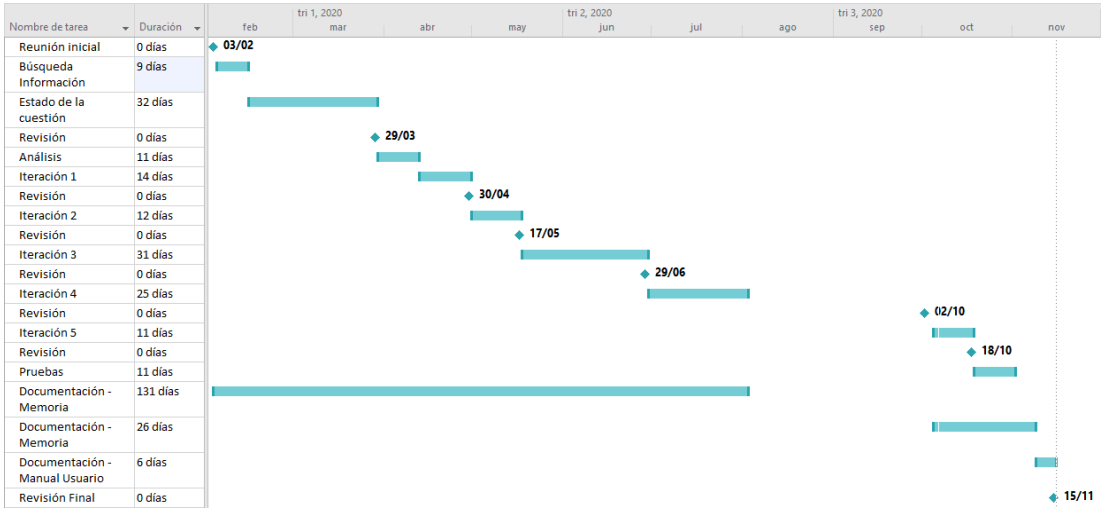


Figura 3.2: Planificación inicial

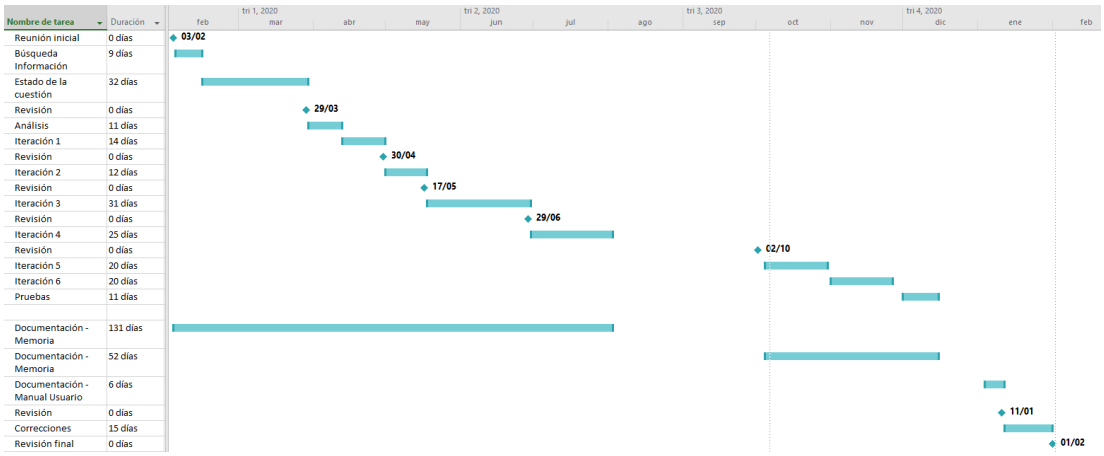


Figura 3.3: Planificación final

Capítulo 4

Desarrollo

En este capítulo se detalla el proceso de desarrollo utilizado para implementar la herramienta. Por una parte, el lenguaje seleccionado para implementar la aplicación es Python, porque es portable (Windows, macOS, Linux), de código abierto y es un lenguaje simple, rápido y sencillo [12]. Por otra parte, el tipo de aplicación desarrollado es de escritorio, de esta forma se consigue evitar comunicaciones con el exterior y por lo tanto obtener una mayor velocidad de procesamiento y manipular los datos de forma más segura. Además, este tipo de aplicaciones suelen ser más robustas y estables [13].

Cada sección de este capítulo se corresponde con una iteración desarrollada. A su vez, cada iteración se dividirá en las siguientes subsecciones:

- Análisis: se analizan los casos de uso a implementar en la iteración actual.
- Prototipo: se diseña un prototipo que cubra los casos de uso resultantes del apartado anterior.
- Implementación: se especifican las bibliotecas empleadas y las decisiones de diseño más relevantes en esta etapa del desarrollo.

4.1 Iteración 1

4.1.1 Análisis

- Añadir caso: se mostrarán los campos que el usuario debe cubrir separados por pestañas, una por cada fase del proceso de peritaje (introducción, adquisición de evidencias, análisis de evidencias, conclusión y anexos).
- Guardar informe: se almacenará el informe en dos formatos (TXT y XML).

4.1.2 Prototipo

En la Figura 4.1 se muestra el prototipo de esta iteración.

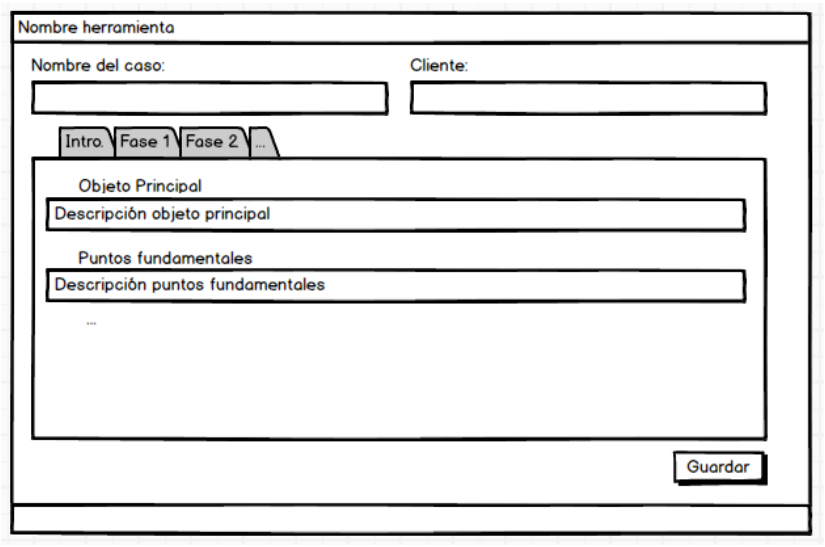


Figura 4.1: Prototipo 1

4.1.3 Implementación

Bibliotecas añadidas:

- Datetime: para establecer la fecha de creación del caso.
- ElementTree: para generar y manipular el archivo XML.

Decisiones de diseño:

- Maquetación: puesto que el número de campos a rellenar por el usuario es muy amplio, para evitar que se deba hacer scroll en la pantalla, se ha decidido mostrar los campos en pestañas separadas dependiendo de la fase del peritaje que se esté tratando (ver Figuras 4.2, 4.3, 4.4, 4.5, 4.6).

  En caso de que el usuario inserte mucha información en los campos de texto, se desplegará una scrollbar para poder visualizar toda la pantalla, ya que en este caso los campos se ampliarán hacia abajo (ver Figura 4.7).

  En cuanto a los colores de la aplicación, se han escogido tonalidades oscuras y de azul para transmitir sensación de tranquilidad, calidad y elegancia, según la psicología de los colores [14]. Para realizar esta maquetación se empleó Gtk.CssProvider, una clase que permite aplicar estilos CSS a los widgets de GTK.

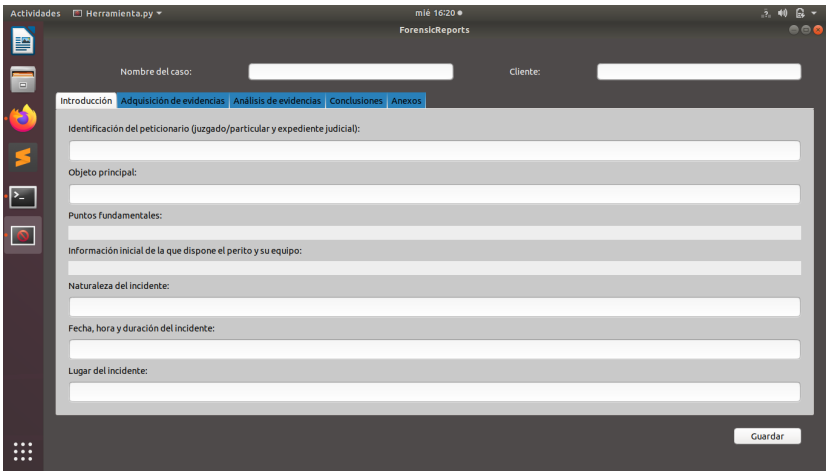


Figura 4.2: Pantalla de creación de un caso - Introducción

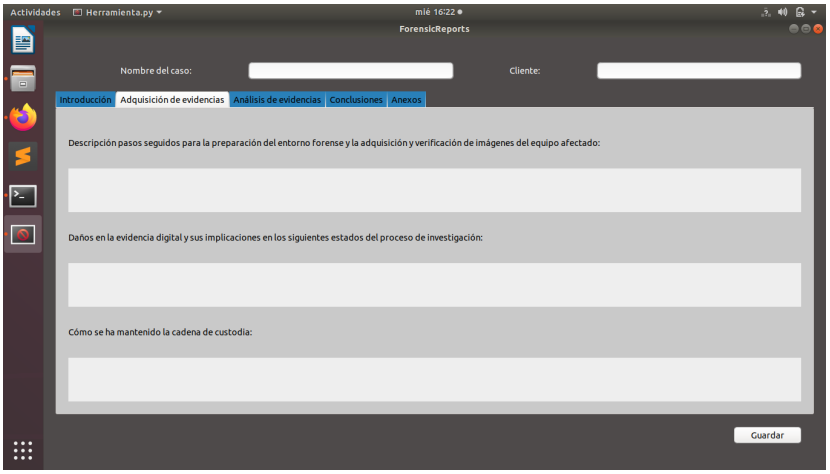


Figura 4.3: Pantalla de creación de un caso - Adquisición de evidencias

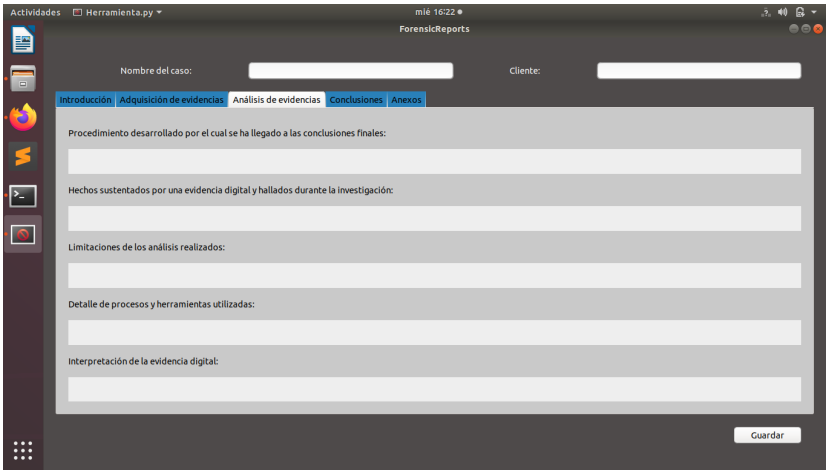


Figura 4.4: Pantalla de creación de un caso - Análisis de evidencias

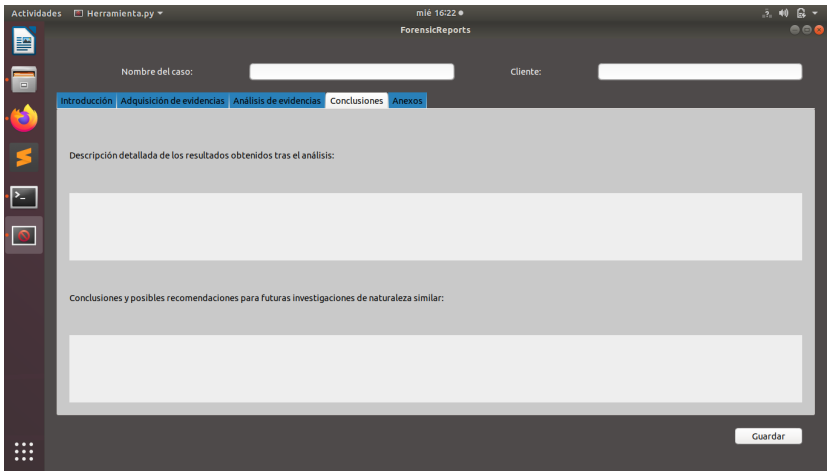


Figura 4.5: Pantalla de creación de un caso - Conclusiones

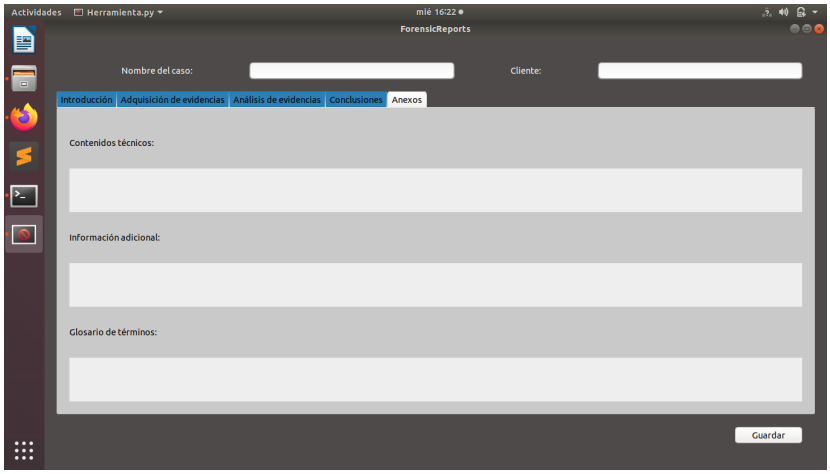


Figura 4.6: Pantalla de creación de un caso - Anexos

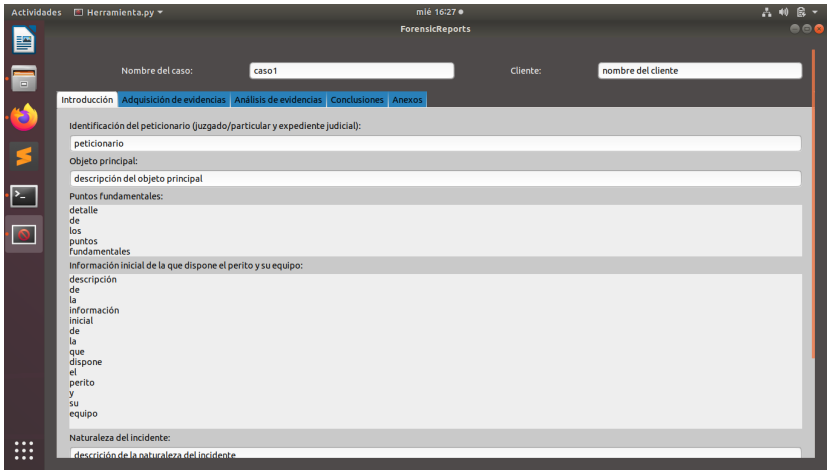


Figura 4.7: Pantalla de creación de un caso - Scrollbar

- Almacenamiento: al guardar el informe se despliega una ventana emergente que permite escoger el directorio en el que almacenar los documentos (ver Figura 4.8). Para organizarlo mejor se creará una carpeta con el nombre del caso introducido en la que se almacenarán dichos documentos (ver Figuras 4.9 y 4.10). Si ya existe un caso con ese nombre se notificará al usuario que está a punto de sobreescribir los datos, dando la opción de aceptar o cancelar. Finalmente, si se termina este procedimiento con éxito, se lanzará un mensaje de aviso (ver Figura 4.11)

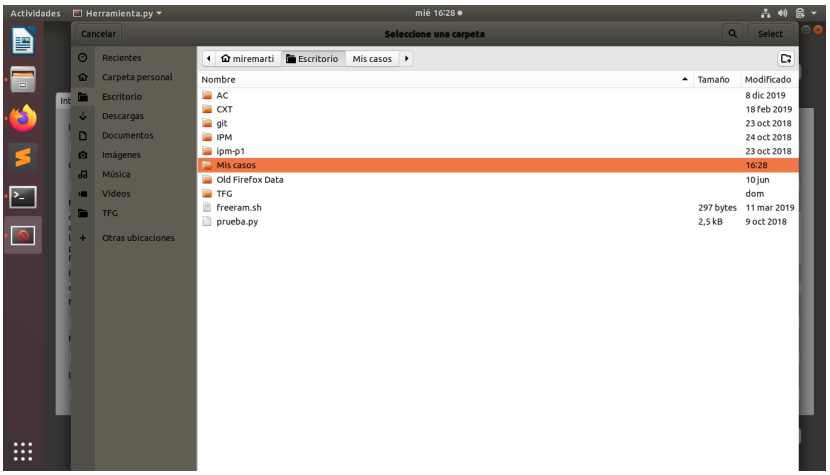


Figura 4.8: Seleccionar ubicación del caso

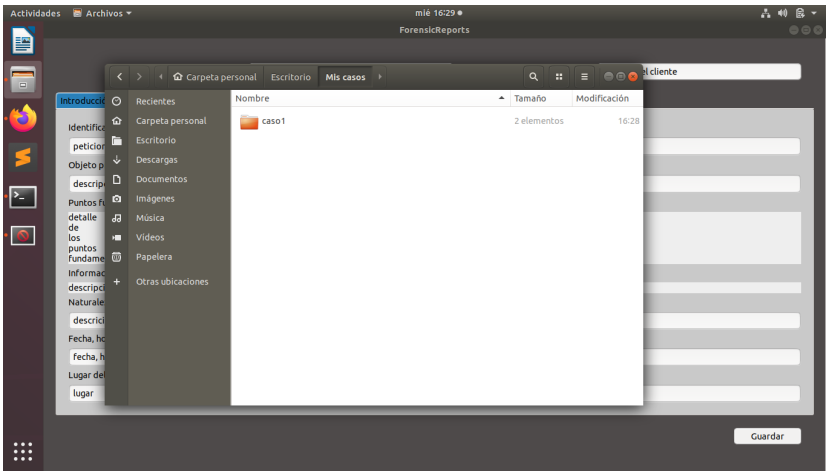


Figura 4.9: Carpeta del caso

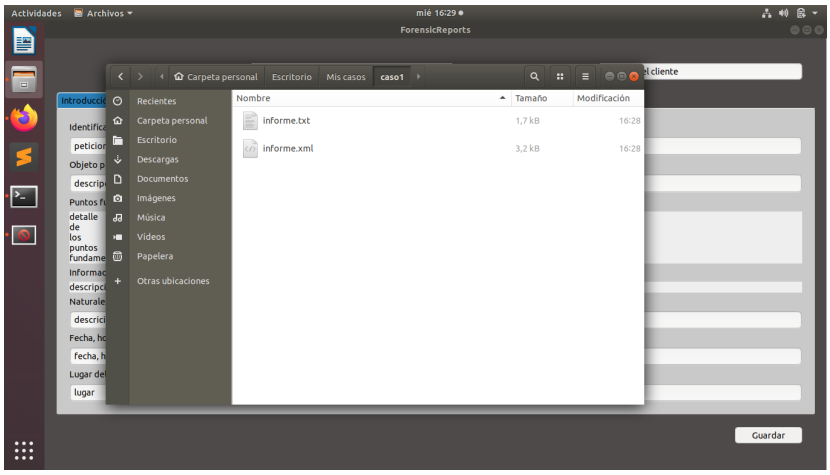


Figura 4.10: Archivos del caso

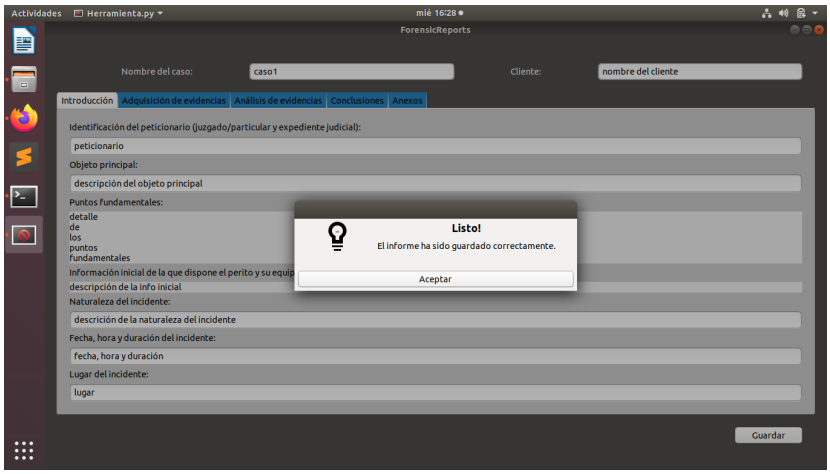


Figura 4.11: Aviso de informe generado correctamente

- Formato del informe: el fin que se persigue con la funcionalidad de guardar el informe es poder abrirlo desde la herramienta para editarlo, o poder exportarlo a otros formatos. Por lo tanto, en esta primera iteración se almacena el informe en formato TXT para visualizarlo de forma más cómoda (ver Figuras 4.12 y 4.13), y en formato XML para manipular los datos fácilmente (ver Figura 4.14), ya que un documento en formato XML es más fácil de procesar [15].

Figura 4.12: Informe en TXT (1)

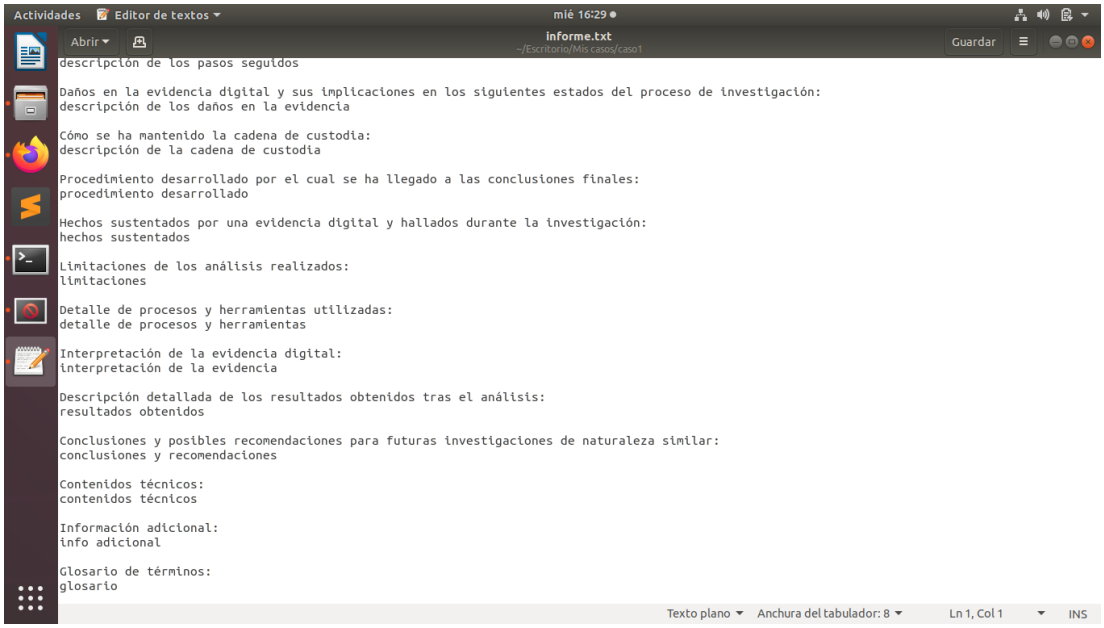


Figura 4.13: Informe en TXT (2)

Figura 4.14: Informe en XML

4.2 Iteración 2

4.2.1 Análisis

- Guardar informe cifrado: se añadirá la función de cifrado al almacenar el documento.
- Editar caso: se podrá modificar la información de un caso que haya sido guardado previamente.

4.2.2 Prototipo

En esta iteración se añade una nueva ventana (ver Figura 4.15) que a partir ahora será la ventana principal de la herramienta. A partir de esta se podrá acceder a la pantalla ya vista en la Figura 4.1.

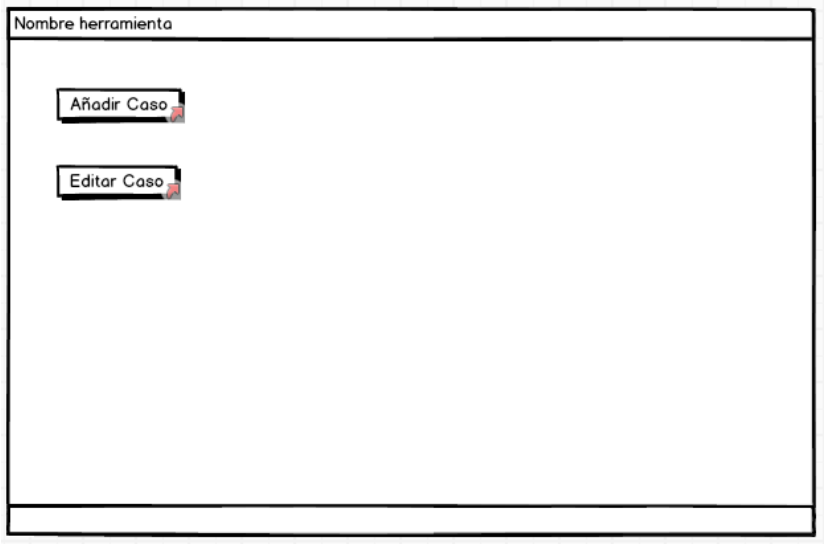


Figura 4.15: Prototipo de la pantalla principal

4.2.3 Implementación

Bibliotecas añadidas:

- Criptography.fernet : para encriptar y desencriptar el informe utilizando AES en modo CBC (ofrece mayor seguridad, sobre todo para el cifrado de imágenes).

Decisiones de diseño:

- Algoritmo de cifrado: dado que el cifrado se empleará para securizar la información almacenada en el propio equipo, se utiliza un algoritmo de cifrado simétrico. Este tipo

de algoritmos ofrecen facilidad de uso y rapidez porque hace uso de una única clave con la que se cifra y descifra el mensaje. En concreto, el algoritmo utilizado es AES (Advanced Encryption Standard) con una clave aleatoria de 128 bits. Que sea aleatoria proporciona mayor seguridad, pero también supone un problema de gestión para el usuario. Este problema se abordará en la tercera iteración.

- Editar caso: se implementó una nueva pantalla que a partir de este momento será la pantalla principal de la herramienta (ver Figura 4.16). Desde ella, se permite añadir o editar un caso. Seleccionando la primera opción se abre la pantalla vista en la Figura 4.2 con los campos vacíos. Eligiendo la segunda opción se abre el explorador de archivos para seleccionar la carpeta del caso que se quiere editar. Una vez seleccionado, se procede a desencriptar el informe y a abrir la pantalla correspondiente para editarlo, con sus campos cubiertos.

  Una vez editado, al guardar se solicita al usuario que seleccione la carpeta en la que desea almacenarlo y, una vez seleccionada, el informe se encripta y se guarda. En caso de que la carpeta sea la misma en la que ya se había almacenado previamente, la aplicación despliega un mensaje de confirmación avisando que se va a sobreescribir el caso (ver Figura 4.17). Si se acepta se procede a sobreescribir el caso, y si se cancela se regresa a la pantalla de edición.

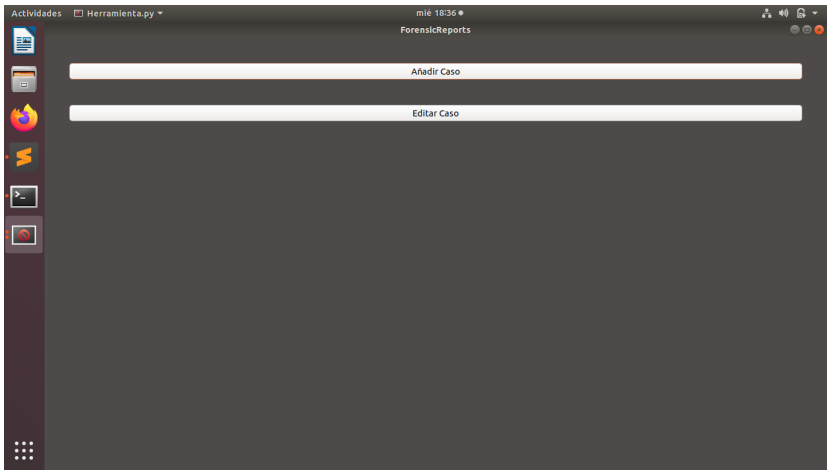


Figura 4.16: Pantalla principal - Iteración 2

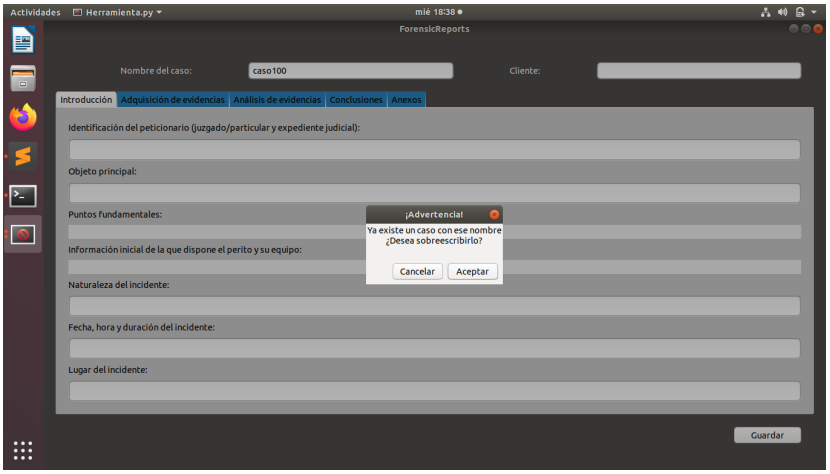


Figura 4.17: Aviso de sobreescritura

4.3 Iteración 3

4.3.1 Análisis

- Autenticación: se le solicitará al usuario una contraseña para acceder a los casos ya almacenados.
- Editar perfil: el usuario tendrá la opción de añadir/editar sus datos personales.

4.3.2 Prototipo

Para el prototipo de esta tercera iteración se incluyen las nuevas pantallas de las Figuras 4.18 y 4.19, y se modifica la pantalla principal tal y como se puede ver en la Figura 4.20.

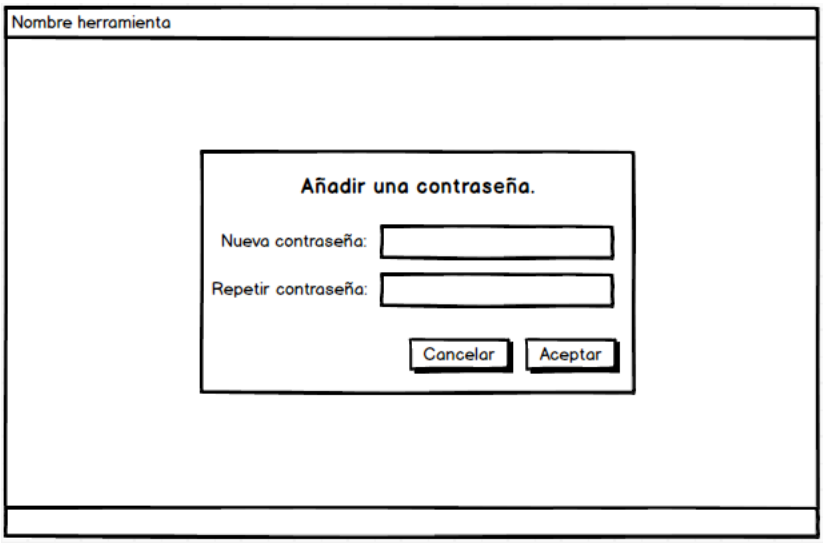


Figura 4.18: Prototipo creación de contraseña

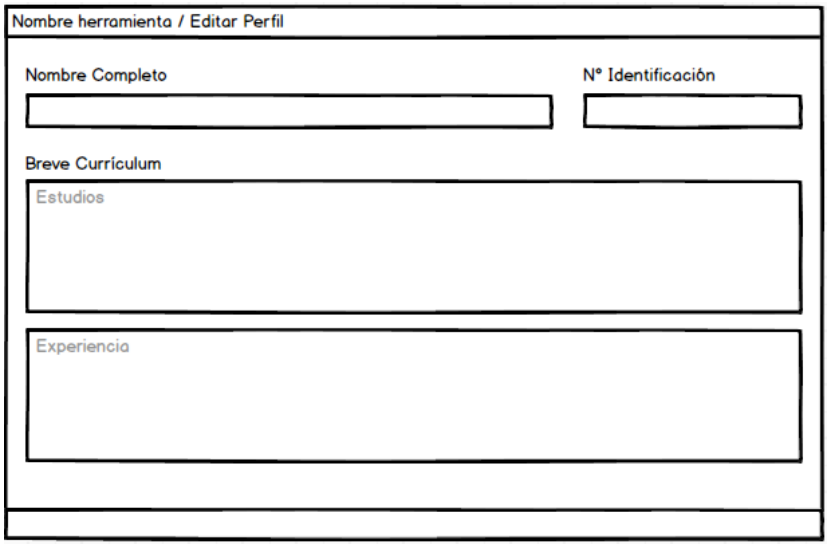


Figura 4.19: Prototipo insertar datos de usuario

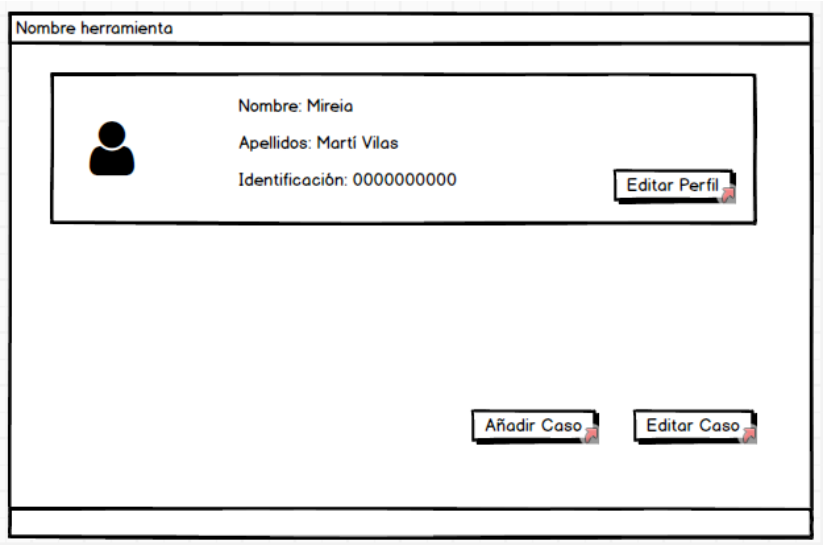


Figura 4.20: Prototipo pantalla principal - Iteración 3

4.3.3 Implementación

Bibliotecas añadidas:

- Crypto: para cifrado asimétrico, generación de claves pública y privada, y su exportación.

Decisiones de diseño:

- Autenticación: el usuario creará una contraseña que será solicitada para poder acceder a los informes. Una vez creada la contraseña se generará un par de claves pública y privada. La pública se almacenará en claro. Sin embargo, para almacenar la privada se utilizará una clave triple DES (derivada de la contraseña insertada por el usuario), para cifrar la clave privada mediante CBC. Este procedimiento solo se llevará a cabo la primera vez que el usuario acceda a la aplicación. Posteriormente, cuando se añada un caso, se generará una clave aleatoria que será utilizada para cifrar el informe con criptografía simétrica utilizando el algoritmo AES (se generará una clave aleatoria para cada caso). Una vez generada, se utilizará la clave pública del usuario para cifrarla utilizando criptografía asimétrica.

  Para abrir un caso y poder editarlo, se realizará el procedimiento inverso. Se solicita la contraseña al usuario, y si es correcta se emplea para obtener la clave privada. Con esta clave privada se descifra la clave simétrica (utilizada previamente para cifrar el informe). Una vez obtenida la clave simétrica, esta se utiliza para descifrar el informe. El esquema descrito se puede ver en la Figura 4.21.

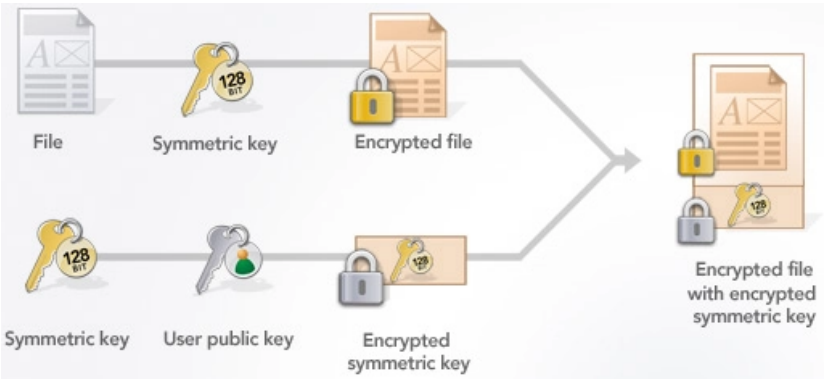


Figura 4.21: Esquema de criptografía híbrida

El procedimiento seguido por este esquema se denomina criptografía híbrida, puesto que se emplea criptografía simétrica y asimétrica para cifrar la información. Es un esquema implementado por herramientas como PGP.

Los pasos seguidos son los siguientes:

- Primer acceso

    1. El usuario ingresa una nueva contraseña.
    2. Generar par de claves pública y privada .

- Almacenar un caso

    1. Generar clave simétrica aleatoria.
    2. Cifrar el informe con la clave simétrica aleatoria utilizando AES (algoritmo de encriptación simétrica).
    3. Cifrar la clave simétrica aleatoria con la clave pública utilizando RSA (algoritmo de encriptación asimétrica).

- Abrir un caso

    1. Solicitar contraseña al usuario.
    2. Obtener la clave privada a partir de la contraseña del usuario.
    3. Descifrar la clave simétrica aleatoria con la clave privada utilizando RSA.
    4. Descifrar el informe con la clave simétrica aleatoria utilizando AES.

En el fragmento de código 4.1 se muestra cómo se genera el par de claves pública y privada. A partir de un número pseudoaleatorio se genera un par de claves pública y privada (de 2048 bits, que ofrece mayor seguridad que la de 1024 bits, y mejor rendimiento que la de 3072 bits). Después, se obtiene la clave pública generada en el paso anterior. Finalmente, se exportan ambas claves (la privada se exporta protegiéndola con la contraseña del usuario), y se almacenan en un archivo .pem.

```
def generate_keys(self, pwd):
    #Genera la clave publica y la privada y las exporta
    #(La privada se exporta con la contraseña del usuario)
    random_generator = Random.new().read
    private_key = RSA.generate(2048, random_generator)
    public_key = private_key.publickey()
    exported_private_key = private_key.exportKey('PEM', pwd, pkcs=1)
    exported_public_key = public_key.exportKey('PEM')

    with open("priv.pem", "wb") as file:
        file.write(exported_private_key)
        file.close()
    with open("pub.pem", "wb") as file:
        file.write(exported_public_key)
        file.close()
```

Listing 4.1: Generar clave pública y privada.

Para comprender mejor los pasos seguidos para cifrar el informe, se puede observar el fragmento de código 4.2 .

```
#GENERACION DEL INFORME Y CIFRADO
nombre_archivo = self.generar_informe_xml(caso, ruta) #Genera el
    informe y devuelve la ruta
clave = Fernet.generate_key() #Genera la clave para cifrado
    simetrico
self.encrypt(nombre_archivo,clave) #Encripta el informe mediante AES
public_key = self.get_public_key() #Obtener clave publica para
    cifrar la clave simetrica
ruta_clave = ruta + '/clave'
self.assimetric_encrypt(ruta_clave, clave, public_key) #Encripta la
    clave simétrica mediante RSA
```

Listing 4.2: Generación del informe y cifrado.

En dicho código se observa que tras generar el informe en formato XML, se crea la clave aleatoria con la que posteriormente se cifra dicho informe. Después, se obtiene la clave pública (previamente generada) y se utiliza para cifrar la clave aleatoria.

Decisiones de diseño (cont.):

- Usuario único: no se considera necesaria la autenticación por usuarios, debido a que la herramienta está pensada para su instalación en equipos personales y ser utilizada por un único perito por equipo.
- Verificación de contraseña: es importante que el usuario introduzca dos veces la nueva contraseña para verificar que no se haya cometido ningún error en su inserción.
- Revocación de contraseña: si el usuario ha olvidado la contraseña deberá crear una nueva dando lugar a generar un nuevo par de claves pública y privada, eliminando así tanto la contraseña anterior como las anteriores claves pública y privada. En consecuencia, no será posible acceder a los informes generados con la clave revocada (ver Figura 4.22).
- Gestión de datos del usuario: durante esta iteración, además de añadir a la pantalla principal la opción de revocar la contraseña, se añade también la opción de editar perfil. Esta opción despliega una nueva pantalla que ofrece la posibilidad de insertar los datos del perito (ver Figura 4.23). Una vez añadidos, estos serán mostrados en la pantalla principal (ver Figura 4.24) y almacenados para adjuntarlos posteriormente a cada informe generado.
- Almacenamiento de datos del usuario: se guardarán en un archivo XML en la carpeta raíz de la herramienta, para facilitar su gestión.

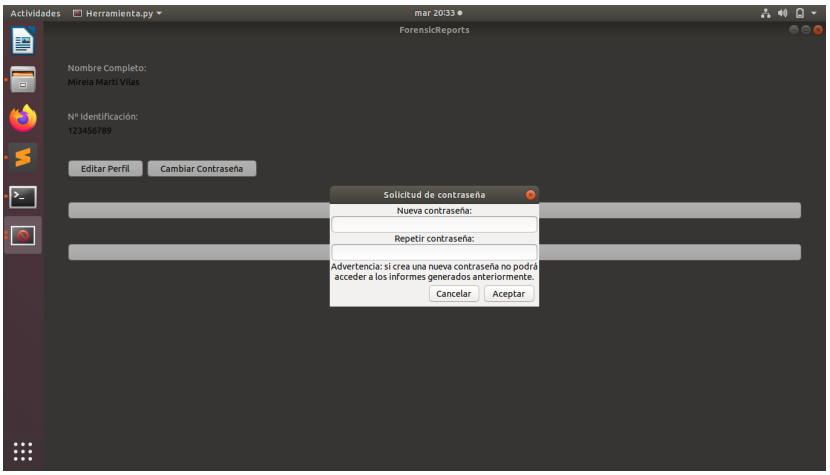


Figura 4.22: Revocar contraseña

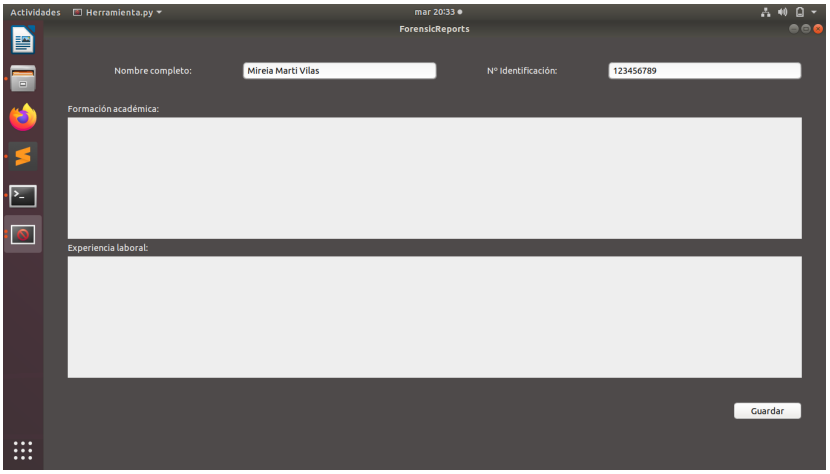


Figura 4.23: Editar datos de usuario

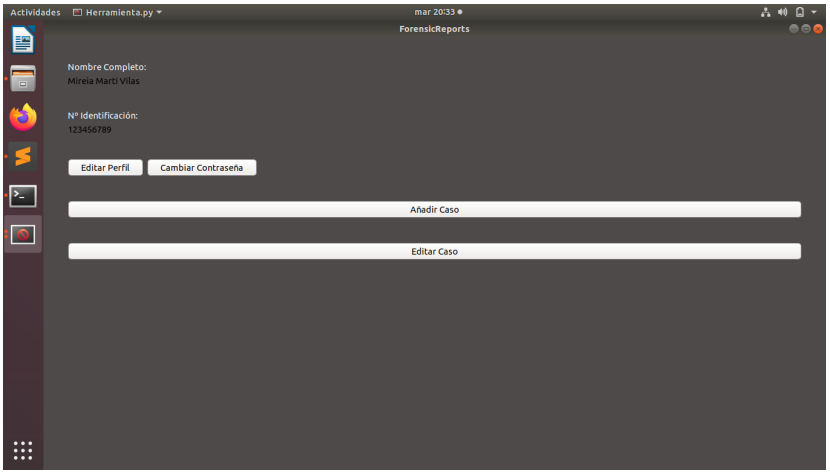


Figura 4.24: Pantalla principal - Iteración 3

4.4 Iteración 4

4.4.1 Análisis

- Añadir evidencia: el usuario podrá adjuntar imágenes para posteriormente referenciarlas en el informe.
- Listar evidencias: se mostrará un listado con las evidencias insertadas.
- Aplicar hash: se aplicará una función hash a cada una de las evidencias para comprobar que no hayan sido modificadas.

4.4.2 Prototipo

Para esta cuarta iteración se modificará la pantalla de edición de casos, tal y como se muestra en la Figura 4.25.

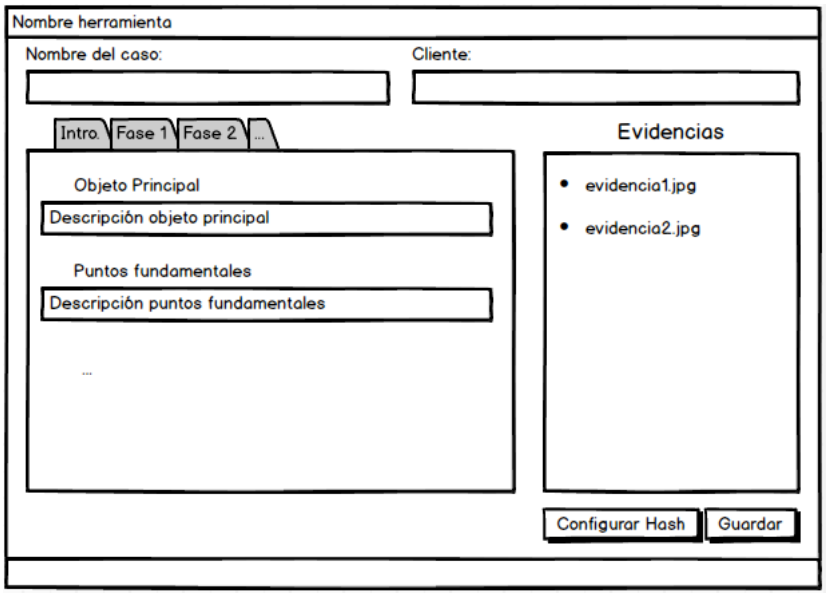


Figura 4.25: Prototipo iteración 4

4.4.3 Implementación

Bibliotecas añadidas:

- Shutil: para mover las evidencias de la carpeta raíz a la carpeta del caso.
- Unidecode: para mostrar los paths de las evidencias aunque tengan caracteres como espacios, acentos, símbolos, etc.

- Hashlib: para aplicar las funciones hash a las evidencias.
- PIL: para manipular imágenes (por ejemplo: comprobar si un archivo es imagen o no).

Decisiones de diseño:

- Adjuntar evidencias: para ofrecer al usuario una forma fácil y cómoda de adjuntar evidencias al caso, se ha implementado un cuadro que permite arrastrar y soltar una o más evidencias (ver Figura 4.26).

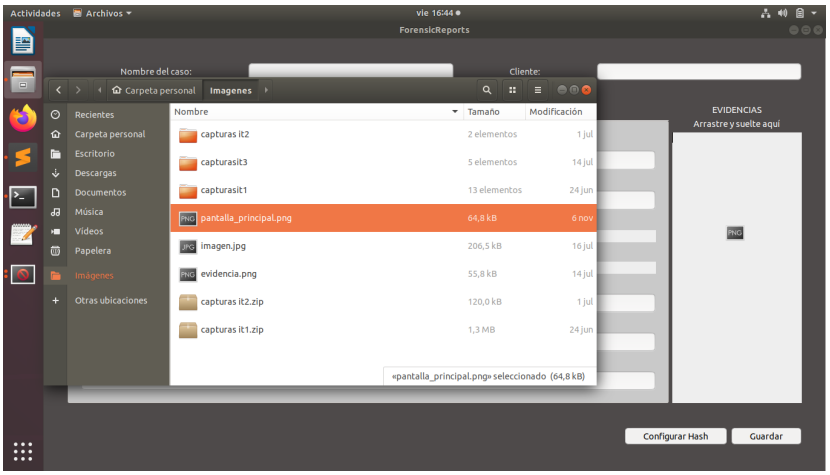


Figura 4.26: Insertar evidencias

- Almacenamiento de evidencias: al soltar las imágenes en el cuadro, estas se añadirán a una nueva carpeta “Evidencias”. Al almacenar el caso, esta carpeta con las evidencias se moverá a la carpeta en la que se creó la carpeta del caso y se cifrarán una a una siguiendo el mismo procedimiento que el explicado para cifrar el informe (ver Sección 4.3.3 - autenticación)
- Mostrar evidencias: para permitir que el usuario referencie las evidencias sin tener que salir de la aplicación, en el mismo cuadro al que se arrastran las imágenes se muestra el nombre de cada evidencia añadida junto con el número de figura para referenciarla fácilmente (ver Figura 4.27). Esta correlación “número de figura - imagen” se almacena en un archivo para asegurar que no se muestren desordenadas al abrir un caso.

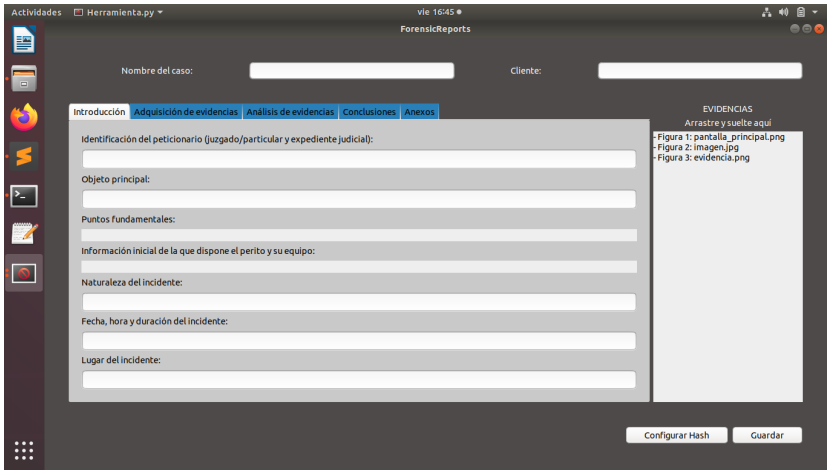


Figura 4.27: Pantalla con evidencias

- Configurar hash: se ofrece al usuario la posibilidad de modificar la función hash que se aplicará a las evidencias (ver Figura 4.28), pudiendo seleccionar uno de los siguientes algoritmos.
  - MD5: función hash con salida de 128 bits. Rápida pero poco segura.
  - SHA-1: parecida a MD5, pero con un resultado de 160 bits.
  - SHA-256: resultado de 256 bits y por lo tanto más lenta que las anteriores pero mucho más segura.

  De forma predeterminada, la aplicación calculará el hash empleando el algoritmo SHA-1, ya que se corresponde con el punto medio entre seguridad y velocidad (es más segura que MD5, y más rápida que SHA-256).

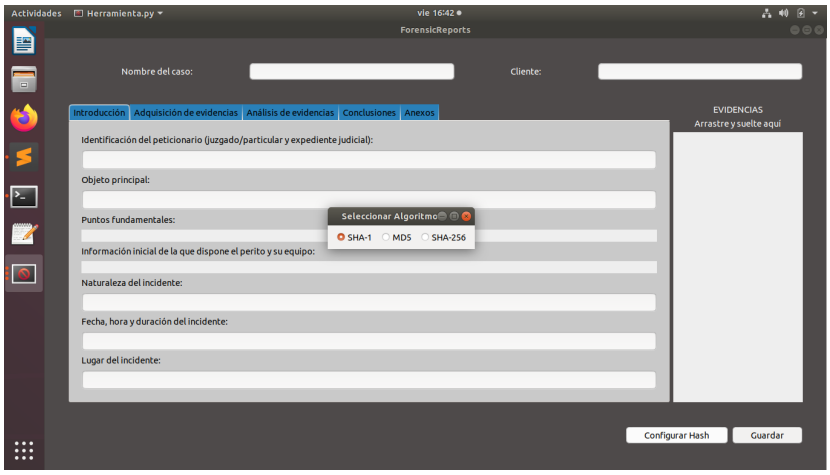


Figura 4.28: Seleccionar función hash

- Aplicar hash: al soltar una imagen en el cuadro de evidencias, se le aplicará una función hash. El resultado de esta función se almacenará en un archivo dentro de la carpeta "Hash" situada en la carpeta del caso, junto con el algoritmo seleccionado. De esta forma, al abrir un caso existente, la herramienta calculará el hash de las imágenes y comparará el resultado con los almacenados en la carpeta "Hash". En caso de que el hash calculado y el almacenado previamente no coincidan, la aplicación lanzará un aviso al usuario (ver Figura 4.29) y procederá a eliminar la evidencia, abriendo el caso con las evidencias restantes (ver Figura 4.30).

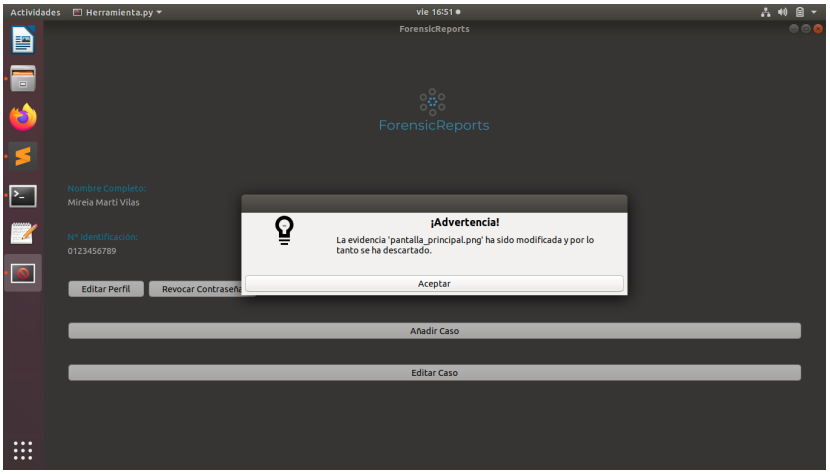


Figura 4.29: Advertencia evidencia modificada

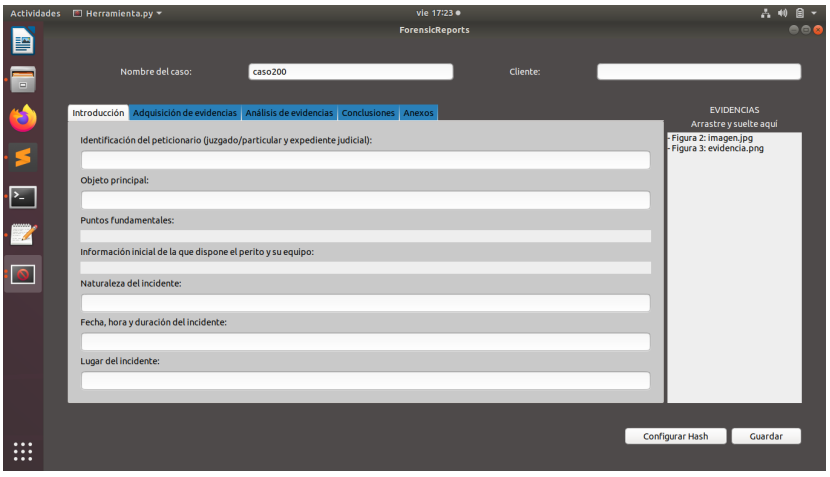


Figura 4.30: Pantalla con evidencia eliminada

4.5 Iteración 5

4.5.1 Análisis

- Exportar a PDF: el usuario podrá exportar a PDF el informe pericial.

4.5.2 Prototipo

En esta quinta iteración se añade un botón al prototipo el cual permite exportar el informe a PDF (ver Figura 4.31).

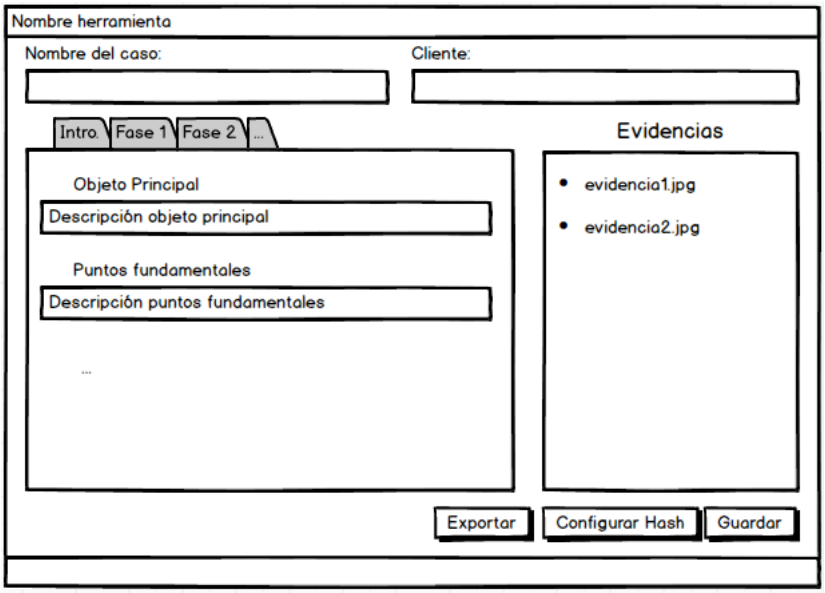


Figura 4.31: Prototipo iteración 5

4.5.3 Implementación

Bibliotecas añadidas:

- Reportlab: para generar y maquetar el informe en PDF.

Decisiones de diseño:

- Generar PDF: el usuario tiene la posibilidad de generar el informe en PDF, ya sea mientras está añadiendo o editando el caso. En el momento en el que pulse el botón para exportar, se generará el PDF y se almacenará el caso del mismo modo que si le diera a guardar.

  El informe en PDF generado se almacenará en la carpeta del caso y no se cifrará a diferencia del resto de la documentación, ya que se da por hecho que el perito desea

exportarlo para compartir o imprimir dicho informe (ver ejemplo de informe PDF en Anexo A).

El archivo consta de los siguientes elementos:

- Título "Informe Pericial".
- Logo de la aplicación ForensicReports.
- Fecha de generación del informe.
- Datos del perito.
- Datos del caso divididos por secciones según las fases del proceso de peritaje (introducción, adquisición de evidencias, análisis de evidencias, conclusiones, anexos y evidencias).

Las evidencias se representan a escala para impedir que la imagen se muestre excesivamente grande. Antes de insertar las imágenes en el archivo PDF, se ordenarán por número de Figura (primero la imagen que se corresponda con la Figura 1, después con la Figura 2, etc...). Esta ordenación se realiza a partir del archivo que contiene la correlación "número de figura - imagen", mencionado anteriormente. Estas imágenes dispondrán de su leyenda correspondiente (Nº Figura - Título Imagen).

4.6 Iteración 6

4.6.1 Análisis

- Accesos recientes: añadir en la pantalla principal un listado con los casos accedidos recientemente para poder acceder a ellos fácilmente.
- Evitar ataque por fuerza bruta: identificar cuando se está tratando de realizar un ataque por fuerza bruta e impedir esta acción.
- Visualizar evidencias: modificar el listado de evidencias para que este permita mostrar una vista previa de las imágenes.

4.6.2 Prototipo

Para esta iteración no fue necesario añadir ningún prototipo.

4.6.3 Implementación

Bibliotecas añadidas:

- En esta iteración no fue necesario añadir ninguna biblioteca.

Decisiones de diseño:

- Accesos recientes: para facilitar el acceso a los casos editados recientemente, se añadió una lista seleccionable de paths. Al guardar un caso, se añade el path al comienzo del archivo "accesos-recientes.txt" (de esta forma aparecerá de primero, pues ha sido el último en ser editado). El límite de paths que se permite almacenar es 15. Si se alcanza este número, se procede a elminar el path más antiguo (es decir, el último en el archivo "accesos-recientes.txt").

  Al seleccionar uno de los elementos de la lista, se solicitará la contraseña para proceder a abrir el caso.

  Como puede verse en la Figura 4.32, el elemento de la lista que ha sido seleccionado se muestra en color azul oscuro y el elemento sobre el que está posicionado el cursor en un azul más claro, para diferenciarlos mejor.

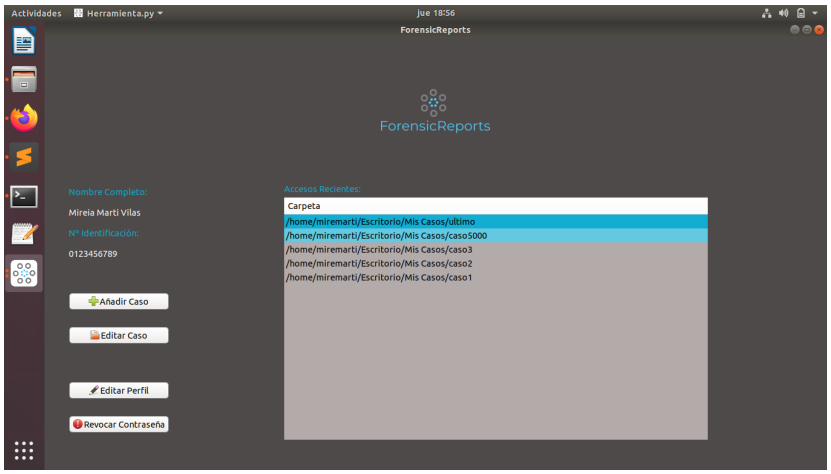


Figura 4.32: Accesos recientes

- Evitar ataque por fuerza bruta: un ataque de fuerza bruta ocurre cuando el atacante emplea determinadas técnicas para probar combinaciones de contraseñas con el objetivo de descubrir las credenciales de una potencial víctima y así lograr acceso a una cuenta o sistema. Existen diferentes tipos de ataque de fuerza bruta, como el "credential stuffing", el ataque de diccionario o el ataque de fuerza bruta inverso [16].

  El "credential stuffing" es un tipo de amenaza que busca robar nuestras credenciales y contraseñas al aprovecharse principalmente de vulnerabilidades en bases de datos [17] , en este caso, al no emplear una base de datos no sería necesario aplicar ninguna medida para solventar este problema.

En los ataques por diccionario, el atacante hace uso de una herramienta que genera una cantidad considerable de palabras y trata de usarlas como contraseña contra un mismo nombre de usuario hasta encontrar la correcta. En un ataque de fuerza bruta inversa se invierte la estrategia de ataque y se comienza con una contraseña conocida, como contraseñas filtradas disponibles en Internet, y con la búsqueda de millones de nombres de usuario hasta encontrar una coincidencia [18].

Existen diversas opciones para impedir que un usuario consiga hacerse con nuestra contraseña mediante el uso de alguna de las dos últimas variantes de ataque por fuerza bruta explicadas en este punto.

Para esta aplicación en concreto se procede a estudiar las dos siguientes soluciones:

- Bloquear la contraseña tras un cierto número de intentos. En este caso, se evita prácticamente por completo un posible ataque por fuerza bruta, ya que tras pocos intentos de insertar la contraseña incorrectamente, esta se invalidaría. Sin embargo, cabría la posibilidad de que el usuario perdiera esta contraseña por error, y con ella el acceso a todos los documentos creados previamente.
- Solicitar al usuario que cree una contraseña robusta. En este caso, la pérdida de contraseña por error no podría darse ya que la contraseña no se bloquearía en ningún momento. Esto conllevaría una mayor probabilidad de éxito durante el ataque que en el caso anterior. Sin embargo, al forzar al usuario a emplear una contraseña robusta (por ejemplo, con mínimo 8 caracteres, entre ellos al menos una mayúscula, una minúscula, un número y un símbolo) se logra que el diccionario empleado por el atacante tenga que ser mucho más elaborado.

Para securizar la aplicación se ha decidido implementar ambas soluciones. Por una parte, se permitirá un máximo de cinco intentos para insertar correctamente la contraseña. De esta manera es poco probable que esta se bloquee por error del usuario. Y por otra parte, se forzará al usuario a crear una contraseña lo suficientemente robusta, pero no tanto como la comentada previamente, ya que crear una contraseña tan elaborada incitaría a anotarla en algún sitio, siendo por lo tanto contraproducente. En este caso se exige que la contraseña tenga por lo menos 6 caracteres con al menos una letra y un número.

Como puede verse en la Figura 4.33, se indica al usuario el número de intentos restantes para ingresar correctamente la contraseña. En caso de que estos intentos se agoten, se solicita al usuario que cree una nueva contraseña (ver Figura 4.34), indicando el formato de la misma, y revocando la contraseña anterior. Esta solicitud de nueva contraseña es la misma que se realiza al arrancar por primera vez la aplicación, o al seleccionar

el botón “Revocar Contraseña”. Si el usuario no cumple con el formato requerido (al menos 6 caracteres incluyendo letras y números), el diálogo volverá a mostrarse hasta que finalmente la contraseña sea válida. Lo mismo ocurre cuando se solicita la contraseña para abrir un caso: mientras el usuario no ingresa la correcta el diálogo seguirá desplegándose.

- Visualizar evidencias: hasta este momento, si el usuario quería visualizar alguna de las evidencias añadidas al caso, tenía que ir a la carpeta correspondiente y abrir la imagen. Para evitar esto, se ha modificado la implementación del listado de evidencias. En vez de un cuadro de texto que muestre las Figuras insertadas, se emplea una lista seleccionable. Para añadir evidencias se emplea el mismo método que hasta ahora: arrastrar y soltar las imágenes sobre el cuadro. Sin embargo, cada uno de los elementos de la lista (que se corresponde con cada una de las Figuras insertadas) serán ahora seleccionables. Cuando el usuario seleccione una de ellas, se abrirá una ventana con la imagen para poder visualizarla mejor y sin necesidad de salir de la aplicación.

  Como puede verse en la Figura 4.35, el elemento de la lista que ha sido seleccionado se muestra en color azul oscuro y el elemento sobre el que está posicionado el cursor en un azul más claro, para diferenciarlos mejor.

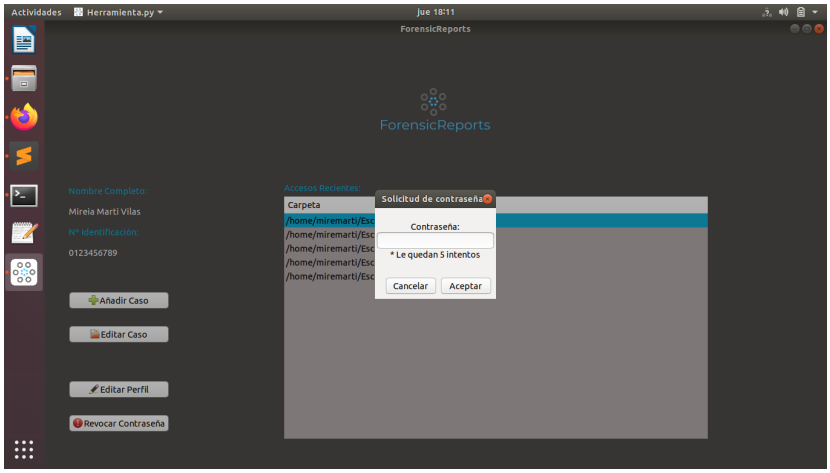


Figura 4.33: Ingreso de contraseña

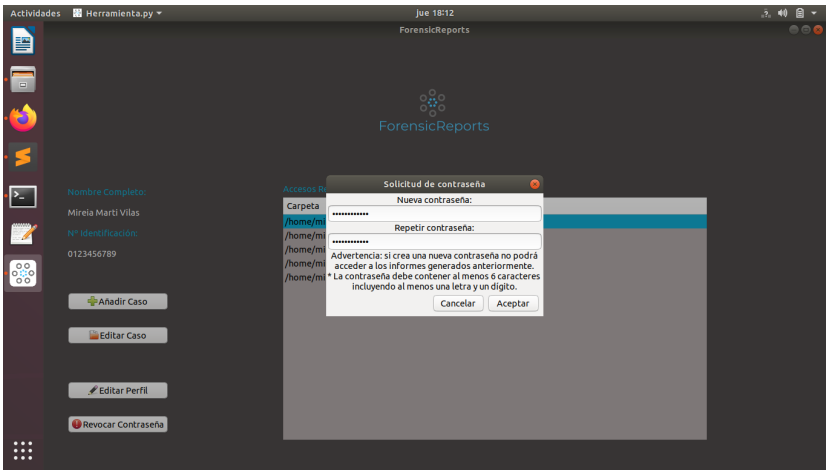


Figura 4.34: Nueva contraseña

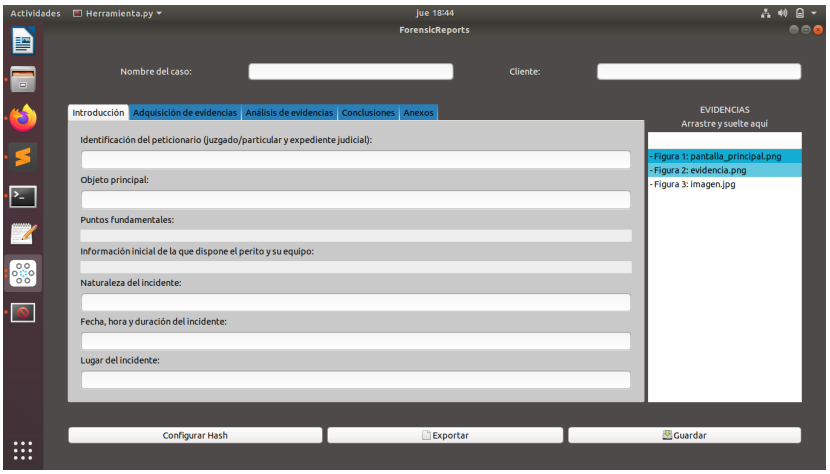


Figura 4.35: Evidencias seleccionables

Capítulo 5

Pruebas

Siguiendo la metodología incremental, tras cada iteración se realizaron pruebas para comprobar que las funcionalidades implementadas funcionaban correctamente.

Por una parte estas pruebas consisten en testear el código, realizando tests sobre los métodos que se consideran más importantes.

Por otra parte, se probó el funcionamiento de la herramienta de forma manual, empleando para ello un caso pericial ficticio.

5.1 Pruebas unitarias

Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección [19]. Existen multitud de herramientas y bibliotecas para realizar estos tests sobre código Python, entre ellas se encuentran DocTest [20], Pytest [21], UnitTest [22] y Coverage [23].

En este caso, y debido a la complejidad de los métodos que se van a testear, emplearemos la biblioteca UnitTest para realizar las pruebas, y la biblioteca Coverage para comprobar la cobertura alcanzada.

5.1.1 Testear funciones hash

En el fragmento de código 5.1 se puede ver cómo se testea el método "aplicar-hash", al que se le pasa el texto al cual se aplicará la función hash, así como el algoritmo a emplear.

```
import unittest
from herramienta import MyWindow, MyPrincipalWindow
from os import remove

class TestPrincipalMethods(unittest.TestCase):
  #aplicar_hash(texto, ruta, algoritmo, nombre)
  #algoritmos disponibles: md5, sha256, sha1
  def test_aplicar_hash_sha1(self):
    hash_esperado = '13237C036448DF7605E2C5BCB378638ECC146FE2'
    hash_calculado = MyWindow.aplicar_hash(self,b'esto es un texto
    de prueba','','sha1','')
    assert hash_esperado.lower() == hash_calculado.lower()

  def test_aplicar_hash_sha256(self):
    hash_esperado =
    '16D011D5F25BC1BCBB3D00944660ACDE8478452B3835E147A6784651332E91F2'
    hash_calculado = MyWindow.aplicar_hash(self,b'esto es un texto
    de prueba','','sha256','')
    assert hash_esperado.lower() == hash_calculado.lower()

  def test_aplicar_hash_md5(self):
    hash_esperado = 'F121D6CE9C2B4CC4FFA7DF6401F59FE7'
    hash_calculado = MyWindow.aplicar_hash(self,b'esto es un texto
    de prueba','','md5','')
    assert hash_esperado.lower() == hash_calculado.lower()
```

Listing 5.1: Testear funciones hash

En el primer método de prueba se testea la función "aplicar-hash" empleando el algoritmo sha1, en el segundo método empleando sha256, y finalmente en el último método empleando md5. La estructura del método de prueba es la misma en los tres casos: a partir de una página web externa [24] se calcula el hash esperado indicando el texto y el algoritmo. Después, se calcula el hash empleando la función "aplicar-hash" implementada. Y finalmente se comparan ambos resultados.

La función "aplicar-hash" tiene dos parámetros opcionales que son la ruta y el nombre. El parámetro ruta indica el path en el que se almacenará el hash obtenido, y el parámetro nombre indica el nombre del archivo en el que se almacenará.

5.1.2 Testear cifrado simétrico

A continuación se detalla cómo se testean los métodos "encrypt" y "decrypt", a los cuales es necesario pasarle el archivo a encriptar/desencriptar y la clave simétrica (que será la misma en ambos casos, ya que el cifrado en este caso es simétrico).

En los tests unitarios es importante generar y destruir los recursos empleados en cada método, para así hacerlos "independientes" entre ellos.

Como se muestra en el fragmento de código 5.2, se comienza generando la clave simétrica y cargándola en la variable "clave". Se genera el archivo que se va a encriptar, se encripta, y se accede a su contenido para realizar la comparación entre el texto original (sin cifrar) y el texto cifrado. En este caso la comparación ha de ser desigual, pues se espera que el texto del archivo haya sido modificado.

```
def test_encrypt_and_decrypt_simmetric(self):
  #generar clave simetrica
  MyWindow.generar_clave(self, 'clave')
  clave = MyWindow.cargar_clave(self,'clave')

  #generar archivo a cifrar
  texto_original = b"Este es un archivo de prueba"
  with open('archivo.txt','wb') as f:
    f.write(texto_original)
    f.close()

  #cifrar - encrypt(self, nombre_archivo, clave)
  MyWindow.encrypt(self, 'archivo.txt', clave)

  #comprobar que el texto del archivo ha sido modificado
  with open('archivo.txt', 'rb') as f:
    texto_cifrado = f.read()
    f.close()
  self.assertNotEqual(texto_original, texto_cifrado)
```

Listing 5.2: Creación de recursos, cifrado y comparación.

Posteriormente, se procede a desencriptar el archivo y, si ambos métodos funcionaron de forma correcta, el contenido del archivo ha de ser (ahora sí) igual que el archivo original, así que se procede a realizar la comparación entre ambas cadenas (ver fragmento de código 5.3).

```
  #descifrar - decrypt(self, nombre_archivo, clave)
  MyWindow.decrypt(self, 'archivo.txt', clave)

  #comprobar que el texto del archivo es igual al original
  with open('archivo.txt', 'rb') as f:
    texto_descifrado = f.read()
    f.close()
  self.assertEqual(texto_original, texto_descifrado)
```

Listing 5.3: Descifrado y comparación.

Finalmente, destruimos los recursos generados, que en este caso fueron el archivo con la clave simétrica, y el archivo con el texto (ver fragmento de código 5.4).

```
#destruir clave y archivo
remove('clave')
remove('archivo.txt')
```

Listing 5.4: Destrucción de recursos

5.1.3 Testear cifrado asimétrico

A continuación se detalla cómo se testearon los siguientes métodos:

- "assimetric_encrypt": empleado para cifrar la clave simétrica con criptografía asimétrica. Es necesario pasarle la ruta completa del archivo en el que se almacenará el resultado, la clave simétrica que se quiere cifrar, y la clave pública que se empleará para realizar el cifrado.

- "decrypt_simmetric_key": empleado para desencriptar la clave simétrica con criptografía asimétrica. Es necesario pasarle la carpeta en la que se encuentra la clave a descifrar y la clave privada

Como se puede observar en el fragmento de código 5.5, se comienza generando una contraseña de usuario (user_password), y generando a partir de ella el par de claves pública y privada. A continuación se genera y se carga la clave simétrica que se pretende cifrar.

```
def test_encrypt_and_decrypt_simmetric_key(self):
    #generar y exportar claves publica(pub.pem) y privada(priv.pem)
    user_password = "hola123"
    MyPrincipalWindow.generate_keys(self,user_password)

    #generar clave simetrica a cifrar
    MyWindow.generar_clave(self, 'clave')
    clave = MyWindow.cargar_clave(self, 'clave')
```

Listing 5.5: Creación de recursos.

El siguiente paso es obtener la clave pública y con ella cifrar la clave simétrica generada anteriormente. Después, se obtiene la clave privada pasándole la contraseña del usuario, y se procede a desencriptar la clave simétrica empleando la clave privada obtenida. Una vez realizado este procedimiento se procede a realizar la comparación entre la clave original y la clave descifrada (ver fragmento de código 5.6).

Finalmente, se destruyen los recursos generados (en este caso la clave simétrica, la clave pública y la clave privada), como puede observarse en el fragmento de código 5.7.

```
    #cifrar
     public_key = MyPrincipalWindow.get_public_key(self)
     MyPrincipalWindow.assimetric_encrypt(self, 'clave', clave,
     public_key)

     #descifrar
     private_key = MyPrincipalWindow.get_private_key(self,
     user_password)
     clave_descifrada =
     MyPrincipalWindow.decrypt_simmetric_key(self, "", private_key)

     #comprobar la clave original es la misma que la descifrada
     self.assertEqual(clave, clave_descifrada)
```

Listing 5.6: Cifrado, descifrado y comparación

```
  remove('clave')
    remove('pub.pem')
    remove('priv.pem')
```

Listing 5.7: Destrucción de recursos.

5.1.4 Comprobando la cobertura

La cobertura del código es un porcentaje que nos permite comprobar la cantidad de código que ha sido testeado. En este caso se ha empleado la biblioteca Coverage, que funciona de la siguiente manera.

Como se puede ver en la Figura 5.1 con el comando "coverage run [archivo test]" podemos ejecutar los tests implementados. En este caso, se han ejecutado cinco métodos test en 0.642 segundos, y el resultado ha sido correcto. A mayores, la biblioteca Coverage ofrece la posibilidad de exportar los resultados del test a un archivo html utilizando el comando "coverage html [archivo del código principal]". Para visualizar este archivo, Python pone a nuestra disposición un comando con el que se levanta un servidor http en localhost: "python3 -m http.server".

```
miremarti@miremarti-VirtualBox:~/Escritorio/TFG4/Testing$ coverage run test.py
.....
----------------------------------------------------------------------
Ran 5 tests in 0.642s

OK
miremarti@miremarti-VirtualBox:~/Escritorio/TFG4/Testing$ coverage html herramienta.py
miremarti@miremarti-VirtualBox:~/Escritorio/TFG4/Testing$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Figura 5.1: Ejecución de pruebas con Coverage

Una vez ejecutados estos comandos, podemos acceder desde el navegador al html generado por Coverage (ver Figura 5.2). Si pinchamos en el enlace con el archivo del código principal

(en este caso, herramienta.py), se accede a una página que contiene los detalles de los tests realizados (ver Figura 5.3). En este caso la cobertura es de un 11% debido a que solo se han realizado los tests de los métodos que se consideran más importantes. Si se quisiera entrar en detalle, pinchando sobre alguno de los módulos de la Figura 5.3 accedemos a una página en la que se muestra el código subrayado en verde si se ha testeado esa línea, o bien en rojo si no se ha testeado.

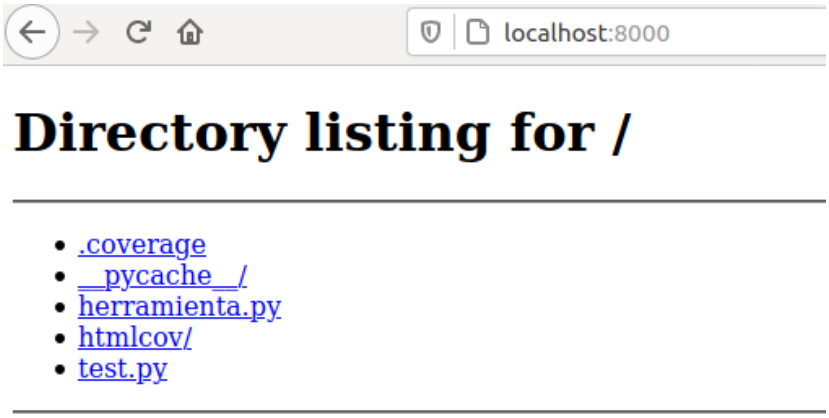


Figura 5.2: Resultados Coverage - Carpeta raíz

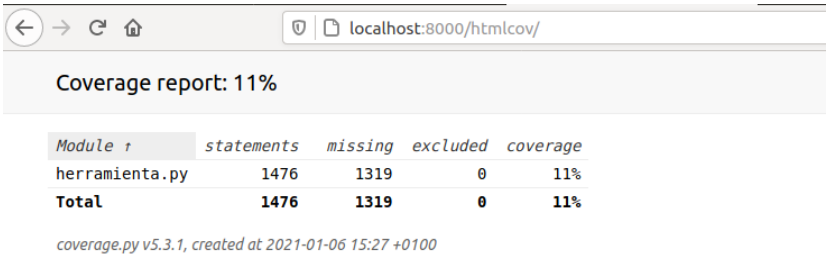


Figura 5.3: Resultados Coverage - Tabla

5.2 Pruebas de interfaz

En este apartado se describen las pruebas realizadas directamente sobre la interfaz de la herramienta. Para ello se ha empleado un ejemplo de un caso pericial ficticio [25].

Una vez insertada la información en los campos correspondientes, se adjuntaron las evidencias del caso y se almacenó. Para comprobar que las evidencias se almacenaron cifradas se intentó acceder a ellas desde el explorador de archivos, obteniendo el resultado visible en la Figura 5.4.

De forma similar, para comprobar que el informe se almacenó cifrado, se intentó acceder al mismo desde el explorador de archivos, obteniendo el resultado visible en la Figura 5.5.

Figura 5.4: Intento de acceso a imagen cifrada

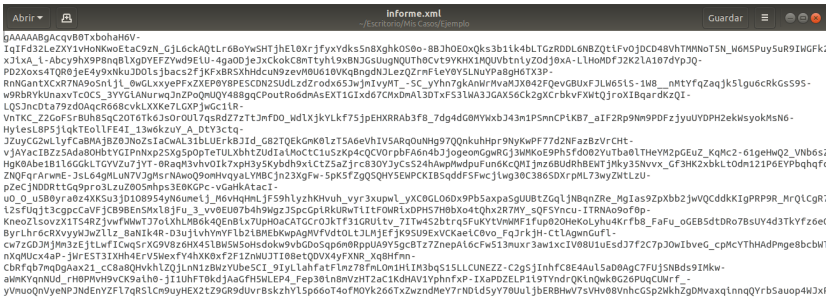


Figura 5.5: Intento de acceso a informe cifrado

Desde el cuadro de accesos recientes se accedió al caso de manera satisfactoria. Luego se editaron los datos de usuario y se comprobó que se actualizaban en la pantalla principal.

Se comprobó también que al modificar alguna de las evidencias desde la carpeta del caso, al tratar de abrirla desde la herramienta esta muestra un aviso indicando que la evidencia fue modificada.

Finalmente, se exportó el informe a PDF (ver Anexo A) y se comprobó que tanto los datos como las evidencias insertadas se muestran correctamente.

Capítulo 6

Conclusiones

En este último capítulo de la memoria se expondrán los resultados obtenidos, así como las conclusiones a destacar y las futuras lineas de desarrollo que podría seguir este proyecto.

6.1 Resultados

Como resultado de este trabajo, y cumpliendo con el objetivo general, se ha obtenido una herramienta para generar informes periciales a partir de los datos introducidos por el usuario.

Uno de los objetivos específicos consistía en facilitar en la medida de lo posible la gestión de la documentación perteneciente a un caso. Este se ha cumplido de manera satisfactoria al permitir al usuario visualizar, almacenar y cifrar las evidencias obtenidas a lo largo del peritaje, de forma fácil y sencilla: solamente hace falta arrastrar las evidencias a la aplicación y guardar el caso.

Además, la información se almacena de forma organizada (almacenando las evidencias en un directorio específico dentro de la carpeta del caso) y en la ubicación que seleccione el usuario, permitiendo así que la localice posteriormente de manera más fácil.

Otro de los objetivos consistía en proteger el acceso a la documentación mediante contraseña. Esto se hace posible gracias al cifrado de la clave privada mediante contraseña, lo cual permite acceder a los informes ya generados sólo si se conoce la misma. Para evitar que la herramienta quede inutilizable si el usuario olvida la contraseña, se proporciona la posibilidad de crear una nueva (eliminando la anterior) para así poder generar informes con una nueva clave.

Con el fin de evitar ataques a dicha contraseña, se ha reducido el número de intentos a cinco para insertarla correctamente. Además, será requerido que dicha contraseña disponga de al menos seis caracteres incluyendo letras y números.

Por otra parte se esperaba proporcionar, de forma sencilla para el usuario, un almacenamiento seguro del informe y la documentación relacionada. Esto se cumple gracias al cifrado

del informe y de las evidencias adjuntas. Este proceso es transparente para el usuario, de forma que si almacena el informe, tanto este como las evidencias se cifran automáticamente tal y como se explicó en la Sección 4.3.3. Además del cifrado, se emplean funciones hash para evitar la manipulación de las evidencias y mantener a salvo la información.

Otro de los objetivos consistía en poder exportar el informe a diferentes formatos una vez cubiertos los campos. En esta versión de la herramienta se ofrece al usuario la posibilidad de exportar el informe a PDF. Sin embargo, el informe también se almacena en formato XML para facilitar la tarea de exportarlo a otros formatos en futuras versiones.

Finalmente, se propuso ofrecer una interfaz clara y sencilla. Para conseguirlo, se distribuyeron los campos en pestañas, evitando así que el usuario tuviera que hacer scroll a lo largo de la pantalla. Cada pestaña incluye los campos que el perito debe introducir en cada una de las fases del peritaje, para proporcionar mayor claridad.

Por otra parte, para que el usuario no introduzca sus datos personales en cada nuevo informe, la herramienta permite insertarlos una única vez, para almacenarlos y emplearlos en cada informe (ofreciendo la posibilidad de editarlos siempre que lo desee el usuario).

Además, la pantalla principal de la herramienta incluye una lista de accesos recientes, para facilitar la labor de editar un caso que haya sido modificado recientemente.

También se insertaron iconos en los botones de la aplicación, para sugerir al usuario la acción que realiza cada uno de ellos. Por último, como ya se ha mencionado al comienzo de este apartado, la interfaz también permite al usuario adjuntar las evidencias de forma rápida, clara y sencilla.

Además de estos objetivos orientados al desarrollo de la herramienta, se propuso distribuir la herramienta como Open Source. Para ello se subió el código a un repositorio público de GitLab[1] y se aplicó una licencia GNU General Public License v3.0 [26]. Esta licencia declara que el software es libre, y lo protege de intentos de apropiación que restrinjan esas libertades a nuevos usuarios cada vez que la obra es distribuida, modificada o ampliada.

Para aclarar posibles cuestiones sobre el funcionamiento de esta aplicación, se ha elaborado un breve manual de usuario que describe las posibles acciones que puede realizar el usuario desde las distintas pantallas de la herramienta (ver Anexo B).

6.2 Conclusiones

La herramienta fue diseñada para proporcionar al usuario una forma de redactar los informes periciales de forma fácil y rápida, sin preocuparse de la seguridad y cumpliendo con los estándares de generación de informes. Todos estos objetivos se cumplieron de forma satisfactoria durante el desarrollo, obteniendo así una herramienta muy útil y fácil de utilizar.

[1] https://gitlab.com/mireia-marti/forensic-reports.git

El uso del lenguaje Python para implementar la aplicación ha sido una muy buena elección. Personalmente, considero que ha sido un lenguaje muy fácil de aprender, rápido y con una gran cantidad de bibliotecas a disposición del programador, lo cual facilitó en gran medida labores como el cifrado de la información. En cuanto a GTK para implementar la interfaz gráfica, considero que es una biblioteca muy potente, pero la documentación existente dificultó en gran medida la implementación de la interfaz, ya que esta es escasa y compleja.

El aprendizaje a lo largo del desarrollo de la herramienta fue continuo. En el aspecto teórico, se invirtió tiempo en comprender la importancia de un informe pericial y cómo debe redactarse, los distintos algoritmos de cifrado y las metodologías de desarrollo existentes. En cuanto al aspecto práctico, se aprendió el lenguaje Python y la biblioteca GTK, además de probar varias herramientas de informática forense para obtener ideas durante la etapa previa al desarrollo.

A lo largo del desarrollo surgieron diversos problemas que fueron solventados poco a poco para cumplir con los objetivos planteados. Entre ellos cabe destacar el problema a la hora de maquetar la aplicación para obtener una interfaz clara y sencilla, y problemas relacionados con el cifrado y la gestión de las evidencias. Sin embargo, al final se logró obtener lo esperado al comienzo de este proyecto: obtener una herramienta para gestionar informes periciales, segura y fácil de utilizar.

6.3 Futuros desarrollos

En este apartado se expondrán distintas ideas que podrían ser implementadas en futuras versiones de la aplicación.

6.3.1 Borrado de evidencias

Hasta este momento la aplicación no ofrece la posibilidad de eliminar una evidencia previamente insertada. El usuario debe acceder a la carpeta del caso y eliminarla de forma manual, y esto resulta tedioso. Una posible mejora sería ofrecer la posibilidad de borrado desde el propio listado de evidencias de la aplicación, por ejemplo, añadiendo esta opción al hacer click derecho sobre algunas de las evidencias.

6.3.2 Archivo solución

Para abrir y editar un caso existente es necesario seleccionar la carpeta del mismo. En caso de que se seleccione otra carpeta o archivo que no se corresponda a la de un caso, se lanza un mensaje de error. Esto puede dar lugar a confusión, pues el usuario podría intentar seleccionar el informe o equivocarse de carpeta. Una posible solución sería restringir la selec-

ción de archivos a los de tipo directorio, pero sigue cabiendo la posibilidad de que el usuario seleccione una carpeta distinta (por ejemplo, la de las evidencias).

Una buena práctica sería generar un archivo solución para cada caso. Este archivo contendría las instrucciones necesarias para abrir el caso, y la herramienta solo permitiría seleccionar este tipo de archivo para su apertura.

6.3.3 Exportar a otros formatos

La herramienta emplea los formatos XML y PDF. El primero para manipular fácilmente la información, y el segundo por comodidad para el usuario. Sin embargo, sería de gran utilidad ofrecer la opción de exportar a otros formatos de texto editable, como por ejemplo ODT, por si en un futuro el usuario necesitara realizar alguna modificación sin necesidad de utilizar la herramienta.

6.3.4 Adjuntar documentación

Hasta este momento es posible adjuntar cualquier tipo de archivo, sin embargo, se ignorarán los que no tengan formato de imagen (.png, .jpg ...). Pero los peritos a menudo emplean otras herramientas de informática forense para, por ejemplo, recolectar evidencias, analizarlas, etc... Estas herramientas ofrecen la posibilidad de extraer informes con los resultados de estos análisis. Por lo tanto, sería de gran utilidad que el perito también pudiera adjuntar esta documentación al caso, de igual manera que puede adjuntar las imágenes.

Apéndices

Apéndice A

Informe Pericial

INFORME PERICIAL

15 de Enero del 2021

Nombre completo:

Mireia Marti Vilas

Nº Identificación:

0123456789

Formación académica:

Grado Ingenieria Informatica por la Universidad de la Coruña

Experiencia laboral:

1. Introducción

Nombre del caso:

Ejemplo

Cliente:

Ramón Ejemplo

Identificación del peticionario (juzgado/particular y expediente judicial):

D. Ramón Ejemplo, 11111111A

Objeto principal:

Nuestro trabajo tiene como finalidad presentar las evidencias digitales de las comunicaciones mantenidas entre el Sr. Ramón Ejemplo y el banco EJEMPLO a través de correo electrónico, y verificar la autenticidad e integridad de los mismos y los archivos adjuntos intercambiados.

Puntos fundamentales:

Obtención de una imagen forense de los ordenadores y buzones de correo de los equipos informáticos de Ramón Ejemplo que contienen las comunicaciones entre el Banco y Ramón Ejemplo y sus familiares.

Reconstrucción de los correos electrónicos que hayan podido ser borrados de los buzones de correo y que sean susceptibles de ser recuperados.

Identificación de la información relevante para el caso y verificación de la autenticidad e integridad de la misma.

Elaboración de un informe pericial detallando el análisis forense realizado sobre las evidencias digitales y su contenido.

Información inicial de la que dispone el perito y su equipo:

Ramón Ejemplo es cliente del área de banca privada del banco EJEMPLO junto con los siguientes miembros de su familia: Ana Ejemplo y María Ejemplo.

El banco EJEMPLO presentó al Sr. Ramón y su familia una oportunidad de inversión en mayo del año 2010.

El Sr. Ramón considera que el Banco no les informó adecuadamente sobre el nivel de riesgo de la inversión y fruto de esta desinformación han sufrido cuantiosas pérdidas económicas.

Naturaleza del incidente:

Desinformación por parte de una entidad financiera.

Fecha, hora y duración del incidente:

N/A

Lugar del incidente:

Madrid

2. Adquisición de evidencias

Descripción pasos seguidos para la preparación del entorno forense y la adquisición y verificación de imá genes del equipo afectado:

La obtención de las imágenes forenses fue realizada siguiendo los procedimientos aplicables de la metodol ogía de trabajo Lazarus, basada en los estándares y mejores prácticas profesionales en tecnología forense y utilizando procedimientos y herramientas específicamente diseñados para llevar acabo este tipo de tra bajos.

Las imágenes obtenidas son firmadas digitalmente con un algoritmo que aplica una función hash sobre el vo lumen completo de las imágenes, de forma que estas no puedan ser manipuladas sin modificar la firma digit al original.

Daños en la evidencia digital y sus implicaciones en los siguientes estados del proceso de investigación:

N/A

Cómo se ha mantenido la cadena de custodia:

ORIGINAL, ORDENADOR personal REEHDD Samsung S/N: S08EJ1MA115070 - Laboratorio Forense de Lazarus

IMAGEN PRIMARIA TARGET, HDD: Western DigitalS/N: WD-WXB1AA023402 - Archivo LFS

IMAGEN SECUNDARIA BACKUP, HDD: Western DigitalS/N: WD-WXB1AA037923 - Archivo LFS

3. Análisis de evidencias

Procedimiento desarrollado por el cual se ha llegado a las conclusiones finales:

Con la ayuda de herramientas especializadas de Tecnología Forense hemos procedido a identificar y extraer de forma automatizada todos los buzones de correo electrónico y copias de seguridad de los mismos almacenados en la imagen forense del ordenador del Sr. Ejemplo. Además, se ha procedido a reconstruir la información que había sido eliminada de los buzones de correo identificados en la imagen forense del ordenador de REE.

Una vez identificados todos los buzones de correos existentes y recuperados los e-mails borrados procedimos a identificar mediante la ayuda de herramientas forenses específicas para el análisis de correo electrónico todas las comunicaciones realizadas con el Banco de EJEMPLO.

Hechos sustentados por una evidencia digital y hallados durante la investigación:

En la Figura 1 se incluye un listado con los buzones de correo identificados.

Hemos realizado una imagen forense de los archivos en formato Word que fueron utilizados para preparar los impresos para la firma de los contratos de la Familia Ejemplo.

Hemos analizados los metadatos existentes en los archivos electrónicos y hemos podido comprobar que las últimas fechas de modificación de dichos archivos han sido el 3 de junio y el 26 y 27 de mayo de 2008.

En la Figura 2 se incluye el listado de los archivos correspondientes a los contratos de la familia Ejemplo y las fechas de creación y modificación incluidas en los metadatos.

En la Figura 3 se incluye un listado (omitido por brevedad) con los correos electronicos intercambiados entre el empleado del Banco Juan y la familia Ejemplo.

Limitaciones de los análisis realizados:

N/A

Detalle de procesos y herramientas utilizadas:

Para el procedimiento de recuperación de información se han utilizado entre otras, las siguientes herramientas forenses: Encase Forensic Edition 6.15 y Nuix.

Tras el análisis forense realizado sobre los buzones de correo identificados en Figura 3, han podido ser recuperados 189 correos electrónicos que habían sido borrados.

Para la redacción del informe pericial se ha empleado ForensicReports, que garantiza una correcta y segura gestión de las evidencias.

Interpretación de la evidencia digital:

Realizamos una imagen forense de todos los correos electrónicos intercambiados entre el empleado del Ban co y la familia Ejemplo.

Dichos correos electrónicos se encontraban en una carpeta denominada “Familia Ejemplo”.

Hemos realizado un análisis de las fechas incluidas en os metadatos de los correos y hemos podido verific ar que estas se corresponden con las fechas de envío y/o recepción de los correos por lo que no existen i ndicios de que estos hayan sido manipulados o alterados.

Así mismo, hemos podido verificar que los correos aportados se corresponden con los registrados en el Log del servidor de correo, por lo que se puede afirmar que los correos aportados conforman la totalidad de las comunicaciones mantenidas entre dicho empleado y los diferentes miembros de la Familia Ejemplo.

4. Conclusiones

Descripción detallada de los resultados obtenidos tras el análisis:

Podemos concluir que no existe ninguna duda acerca de la integridad y autenticidad de los correos y que h emos verificado que todos ellos han sido enviados desde la cuenta de correo de REE y no existe ningún ind icio de que estos puedan haber sido manipulados.

Conclusiones y posibles recomendaciones para futuras investigaciones de naturaleza similar:

Hemos realizado un análisis forense sobre los correos seleccionados y no existe ninguna duda sobre su int egridad y autenticidad. Por lo que constituyen a nuestro juicio una copia fiel de la información original

.

5. Anexos

Contenidos técnicos:

Información adicional:

Glosario de términos:

6. Figuras

Buzón de correo	Ubicación
Archive.pst	D:\Exportados\EmailBruto\Administrador\archive.pst
Backup.pst	D:\Exportados\EmailBruto\Administrador\backup.pst
outlook.pst	D:\Exportados\EmailBruto\Administrador\Outlook.pst
outlook.pst	D:\Exportados\EmailBruto\JOSEHDantiguo\outlook.pst

Figura 1: buzones_correo.png

Name	Content Last Modified	Content Created	File Last Modified
MARIA EJEMPLOEJEMPLO_36245.doc	03/06/2008 8:56	27/05/2008 10:54	03/06/2008 8:56
MARIA EJEMPLOEJEMPLO_36258.doc	27/05/2008 9:12	26/05/2008 16:37	27/05/2008 9:12
JOSE EJEMPLOEJEMPLO_36245.doc	03/06/2008 10:24	03/06/2008 10:15	03/06/2008 10:24
ANA MARIA EJEMPLO_36245.doc	03/06/2008 10:15	03/06/2008 10:11	03/06/2008 10:15
ANA MARIA EJEMPLO_36258.doc	26/05/2008 16:50	26/05/2008 16:20	26/05/2008 16:50
ANA MARIA EJEMPLO_36245.doc	03/06/2008 10:10	03/06/2008 9:29	03/06/2008 10:10
ANA MARIA EJEMPLO _36258.doc	26/05/2008 16:53	26/05/2008 16:52	26/05/2008 16:53

Figura 2: contratos.png

Title / Subject	Content Last Modified	Content Created	Sent	Received
	15/07/2010	15/07/2010	15/07/2010	15/07/2010
RE: anulación plazo fijo	8:50	8:48	8:50	8:50
RV: NO REENEJEMPLOR: Posible	29/08/2008	29/08/2008	29/08/2008	29/08/2008
Cancelación Depos Estructurados	12:19	11:40	11:41	11:41

Figura 3: correos_electronicos.png

Apéndice B

Manual de usuario

Forensic Reports

Manual de usuario

Mireia Martí Vilas

Enero 2021

Índice

1. Introducción

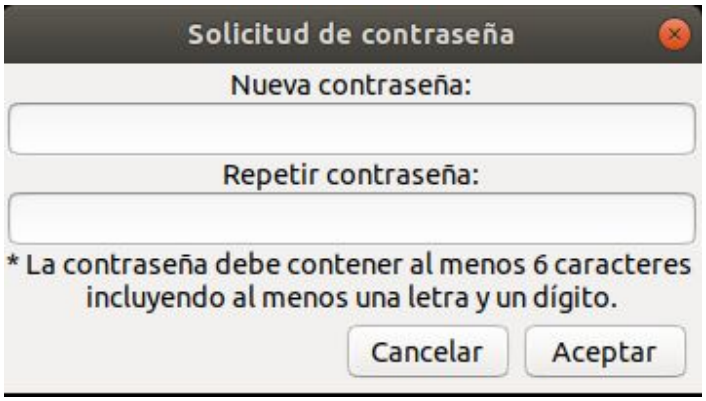
El presente documento consiste en un breve manual de usuario que expondrá el funcionamiento de la herramienta ForensicReports.

Esta herramienta está pensada para facilitar la labor de elaborar informes periciales y gestionar la documentación relacionada con el caso a la vez que se mantiene a salvo la información.

Este manual está estructurado por secciones. Cada sección detallará la información mostrada en cada una de las pantallas de la herramienta, y describirá las funciones que puede realizar el usuario desde las mismas.

2. Primer acceso

La primera vez que el usuario arranque la herramienta, aparecerá la siguiente ventana.

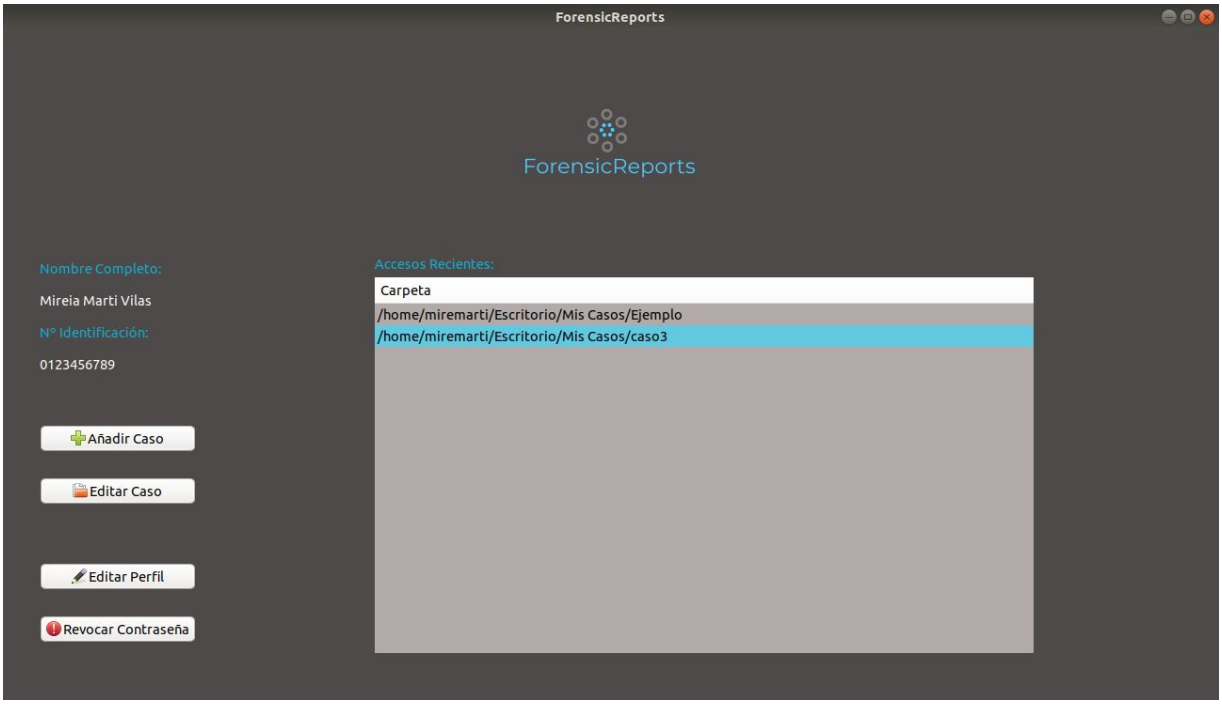


El usuario deberá insertar la contraseña que posteriormente será solicitada para acceder a los casos almacenados.

Esta contraseña debe cumplir los requisitos especificados: mínimo 6 caracteres incluyendo caracteres y mínimo un dígito.

3. Pantalla principal

En la siguiente imagen se muestra la pantalla principal de la herramienta.



Durante el primer acceso de la aplicación, si se ha creado correctamente la contraseña de usuario, se accede a esta pantalla.

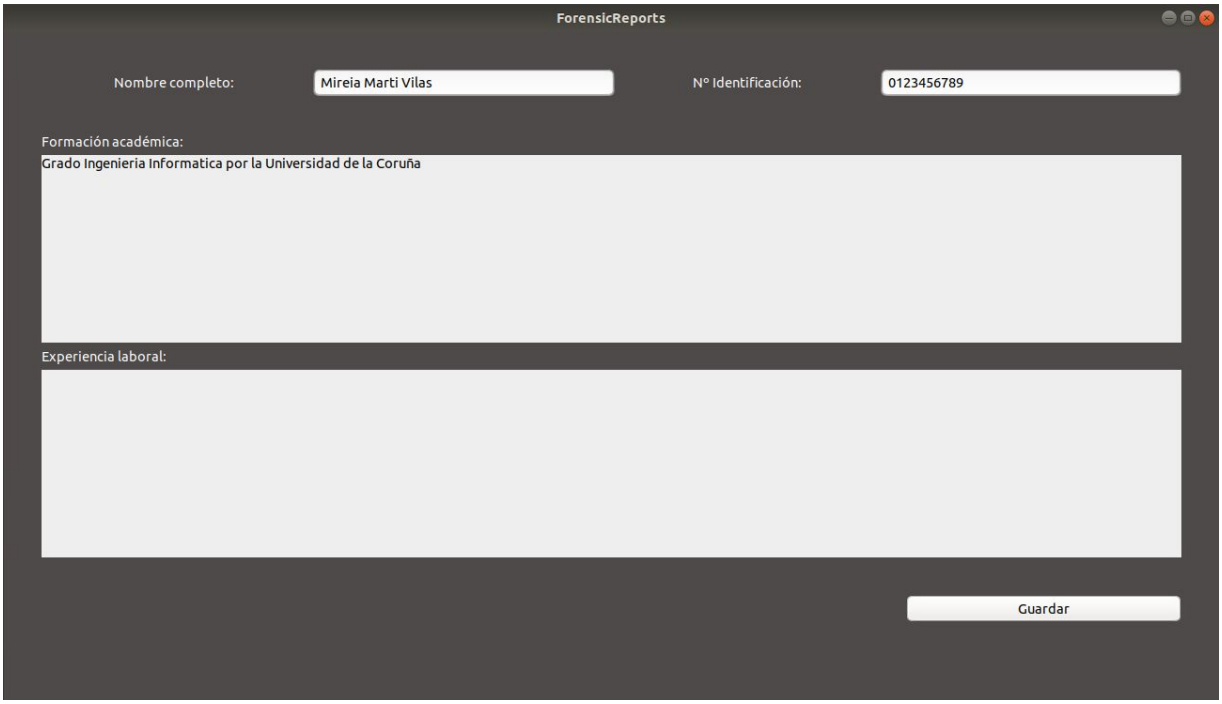
Si no es el primer acceso se accederá a esta pantalla directamente.

Posibles acciones

1. Añadir caso: se abrirá una nueva pantalla que permite añadir un caso (apartado 5).
2. Editar caso: se abrirá el explorador de archivos para seleccionar el caso a editar (apartado 6).
3. Editar perfil: se abrirá una pantalla que permite añadir/editar los datos de usuario (apartado 4).
4. Revocar contraseña: permite eliminar la contraseña actual y crear una nueva (apartado 7).
5. Accesos recientes: permite acceder a un caso modificado recientemente (apartado 6).

4. Datos de usuario

Al seleccionar esta opción se despliega una nueva ventana mostrada en la siguiente imagen.



Esta ventana permite insertar los datos del perito. Estos datos serán añadidos a cada informe exportado.

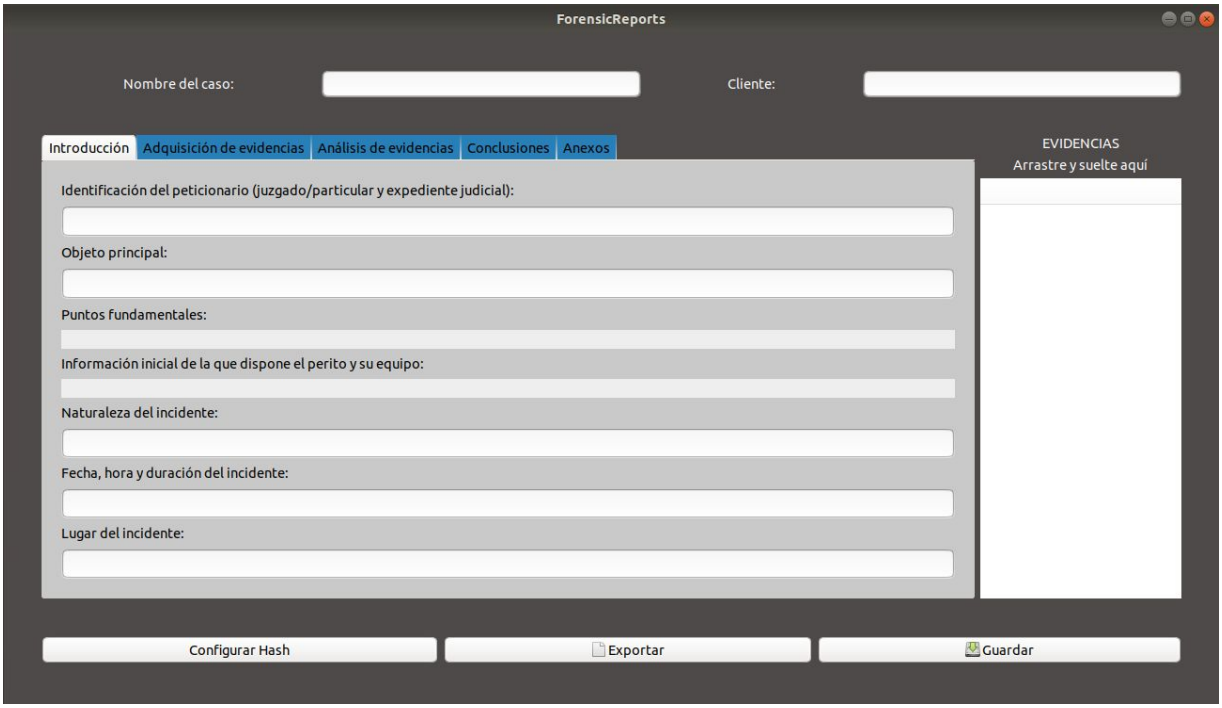
Posibles acciones

1. Guardar: se almacenarán los datos de usuario y se cerrará la ventana.

Una vez insertados los datos nombre y número de identificación, al guardar los cambios, estos se mostrarán en la pantalla principal.

5. Añadir caso

Al seleccionar esta opción se despliega una nueva ventana mostrada en la siguiente imagen.



Esta ventana permite insertar los datos del caso y adjuntar las imágenes relacionadas.

El campo “Nombre del caso”, será el empleado para nombrar la carpeta en la que se almacenará toda la información relacionada con el caso.

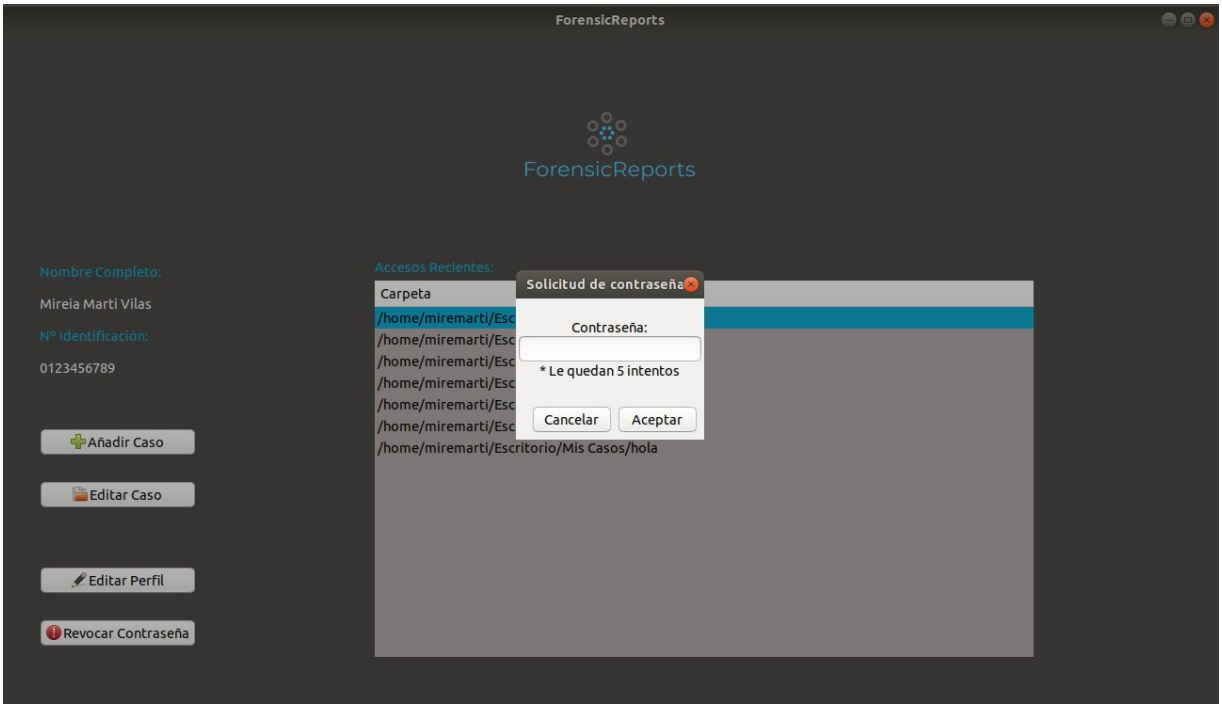
Posibles acciones

1. Adjuntar evidencias: arrastrando y soltando imágenes al cuadro de evidencias, se almacenarán y se mostrarán en el listado (acompañadas del número de figura correspondiente).
2. Configurar hash: se mostrará una ventana emergente que permite seleccionar la función hash que se aplicará a las evidencias.
3. Exportar: se almacenará el caso, se exportará a PDF (almacenándolo en la carpeta del caso) y se cerrará la ventana.
4. Guardar: se mostrará el explorador de archivos para que el usuario seleccione la carpeta en la que se quiere almacenar el caso. Una vez seleccionada la ruta, el caso se almacenará y se cerrará la ventana.

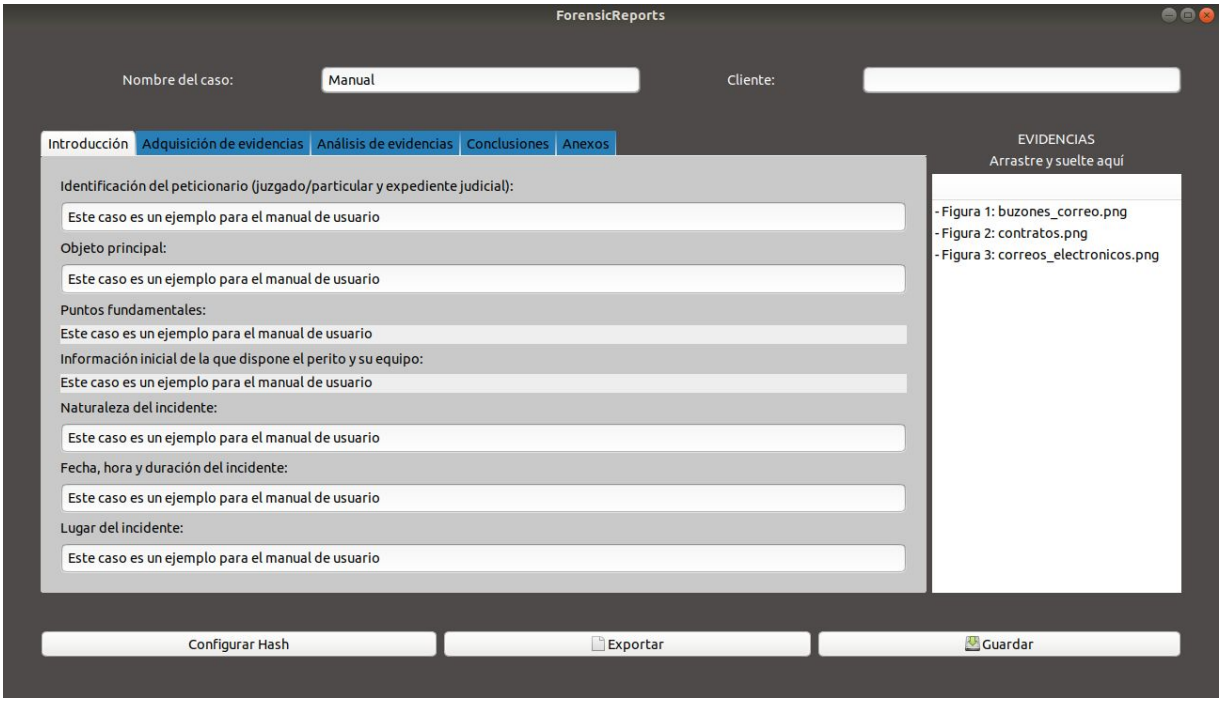
6. Editar caso

Para editar un caso existen dos opciones. La primera es seleccionar el caso desde el cuadro de accesos recientes que se muestra en la pantalla principal. La segunda opción es seleccionar el botón “Editar Caso”, que abrirá el explorador de archivos para permitir seleccionar la carpeta del caso que se desee editar.

Una vez seleccionado el caso, se desplegará una ventana emergente como la mostrada a continuación, para que el usuario inserte su contraseña. Mientras esta no sea la correcta, el diálogo seguirá mostrándose.



Una vez insertada la contraseña correctamente, se abrirá una nueva ventana con los datos del caso rellenados, tal y como se muestra en la siguiente imagen.

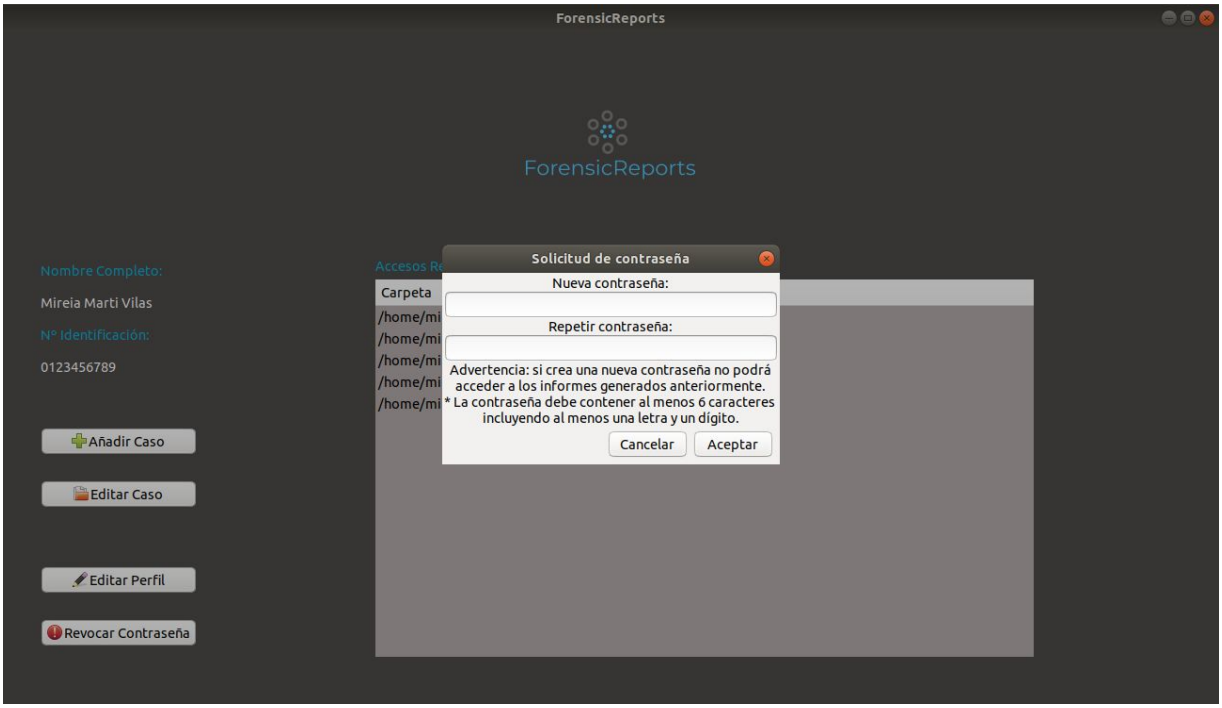


Posibles acciones

1. Adjuntar evidencias: arrastrando y soltando imágenes al cuadro de evidencias, se almacenarán y se mostrarán en el listado (acompañadas del número de figura correspondiente).
2. Configurar hash: se mostrará una ventana emergente que permite seleccionar la función hash que se aplicará a las evidencias.
3. Exportar: se almacenará el caso, se exportará a PDF (almacenándolo en la carpeta del caso) y se cerrará la ventana.
4. Guardar: el caso se sobreescribirá y se cerrará la ventana.

7. Revocar contraseña

En caso de que el usuario desee crear una nueva contraseña, deberá seleccionar esta opción. Al hacerlo se abrirá una ventana emergente como la mostrada en la siguiente imagen.



Tal y como se advierte en el diálogo, es importante destacar que al revocar la contraseña el usuario no podrá acceder a los casos que haya almacenado hasta el momento.

Posibles acciones

1. Aceptar: si la contraseña insertada cumple con los requisitos (mínimo 6 caracteres incluyendo letras y números), se eliminará la contraseña anterior, se almacenará la nueva y se cerrará la ventana emergente.
2. Cancelar: se cerrará la ventana emergente descartando los cambios.

Lista de acrónimos

AES Advanced Encryption Standard. 39, 44, 45

CBC Cipher Block Chaining. 39, 44

DES Data Encryption Standard. 44

HPA Host Protected Area. 14

IOCE International Organization of Computer Evidence. 1

KML Keyhole Markup Language. 10

PGP Pretty Good Privacy. 44

RAM Random Access Memory. 5

SO Sistema Operativo. 2

TFG Trabajo Fin de Grado. 1, 2

Bibliografía

[1] P. V. Avendaño, *Técnicas de Análisis Forense informático para Peritos Judiciales profesionales.* 0xWord, 2018.

[2] "Página web oficial osforensics." [En línea]. Disponible en: https://www.osforensics.com/

[3] "Página web oficial prodiscover." [En línea]. Disponible en: https://www.prodiscover.com/

[4] "Página web oficial encase." [En línea]. Disponible en: https://www.guidancesoftware.com/encase-forensic

[5] "Página web oficial forensicexplorer." [En línea]. Disponible en: http://www.forensicexplorer.com/

[6] "Página web oficial belkasoft." [En línea]. Disponible en: https://belkasoft.com/x

[7] "Página web oficial cellebrite." [En línea]. Disponible en: https://www.cellebrite.com/es/pagina-principal/

[8] "Página web oficial autopsy." [En línea]. Disponible en: https://www.autopsy.com/

[9] AENOR, "Une 50132:1994 numeración de las divisiones y subdivisiones en los documentos escritos." [En línea]. Disponible en: http://etitudela.com/fpm/gdsa/downloads/une50132iso2145.pdf

[10] A. L. F. Galea, "Importancia del método de la prueba pericial en materias de tecnología y su impacto en la agilidad del proceso judicial," 2019. [En línea]. Disponible en: https://www.acta.es/medios/articulos/cultura_y_sociedad/059001.pdf

[11] J. M. T. Cano, "Metodología para el desarrollo de procedimientos periciales en el ámbito de la informática forense." [En línea]. Disponible en: https://ruidera.uclm.es/xmlui/bitstream/handle/10578/6667/TFG_Juan_Miguel_Tocados.pdf?sequence=1&isAllowed=y

[12] "Ventajas y desventajas — materiales del entrenamiento de programación en python - nivel básico." [En línea]. Disponible en: https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/ventajas_desventajas.html

[13] "¿web o desktop? principales diferencias entre aplicaciones web y desktop - ventajas y desventajas de aplicaciones web y desktop." [En línea]. Disponible en: http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones-web-y-aplicaciones-desktop

[14] G. Mier, "¿qué colores y tipografías elijo para el diseño de mi app?" 2017. [En línea]. Disponible en: https://pickaso.com/2017/consejos-color-tipografia-apps

[15] D. P. Fernandez, "¿qué es xml? ventajas y desventajas." 2018. [En línea]. Disponible en: https://tecnonucleous.com/2018/11/09/que-es-xml/

[16] "Qué es un ataque de fuerza bruta y cómo funciona," 2020. [En línea]. Disponible en: https://www.welivesecurity.com/la-es/2020/06/24/que-es-ataque-fuerza-bruta-como-funciona/

[17] J. Jiménez, "Qué es credential stuffing y cómo protegernos de esta amenaza," 2020. [En línea]. Disponible en: https://www.redeszone.net/tutoriales/seguridad/que-es-credential-stuffing/

[18] Kaspersky, "¿qué es un ataque de fuerza bruta?" 2018. [En línea]. Disponible en: https://latam.kaspersky.com/resource-center/definitions/brute-force-attack

[19] Yeeply, "¿qué son las pruebas unitarias y cómo llevar una a cabo?" 2019. [En línea]. Disponible en: https://www.yeeply.com/blog/que-son-pruebas-unitarias/

[20] P. S. Foundation, "doctest — test interactive python examples." [En línea]. Disponible en: https://docs.python.org/3/library/doctest.html

[21] P. org, "pytest: helps you write better programs." [En línea]. Disponible en: https://docs.pytest.org/en/stable/

[22] P. S. Foundation, "unittest — unit testing framework." [En línea]. Disponible en: https://docs.python.org/3/library/unittest.htmlhttps://docs.python.org/3/library/unittest.html

[23] R. T. Docs, "Coverage.py 5.3.1 documentation." [En línea]. Disponible en: https://www.google.com/search?q=coverage+library&oq=coverage+library&aqs=chrome..69i57j0i19j0i19i22i30l6.3615j0j4&sourceid=chrome&ie=UTF-8

[24] P. LLC, "Online hash calculator." [En línea]. Disponible en: https://www.pelock.com/products/hash-calculator

[25] J. L. G. Gómez, "Informe sobre el peritaje informático." [En línea]. Disponible en: http://www.institutopascualmadoz.es/wp-content/uploads/2016/06/TFM-Jos%C3%A9-Luis-Garc%C3%ADa-G%C3%B3mez.pdf

[26] F. S. Foundation, "Licencias gnu." [En línea]. Disponible en: https://www.gnu.org/licenses/licenses.es.html