Universidad de Valladolid

Escuela de Ingeniería Informática de Valladolid

Trabajo de Fin de Grado

Grado de Ingeniería Informática

Sleuth Assistant

Autor:

Juan Miguel Celorrio de la Torre

Tutor:

Joaquín Adiego Rodríguez

Índice general

Índice de tablas

Índice de códigos

Índice de figuras

Agradecimientos

Se muestra agradecimiento a todos los profesores de la universidad que tanto se han preocupado no solo de mejorar la experiencia de nuestro paso por la universidad, si no la preocupación que han mostrado por nuestro aprendizaje y por infundinos valor para buscar cosas más alla del temario por nosotros mismos y no limitarnos a lo que los libros pueden ofrecernos en una buena lectura, si no en probar todas esas cosas por nosotros mismos.

También me gustaría mostrar un pequeño agradecimiento a mi hermano pequeño por sacar incontables fallos a cada uno de mis programas y por consecuencia haberme dado la posibilidad de mejorarlos.

Capítulo 1

Introducción y Objetivos

Este trabajo surge como ampliación de los conocimientos adquiridos en la asignatura informática forense, en concreto sobre las técnicas y buenas maneras de tratamiento de ficheros, metadatos y aún más en concreto con la identificación de ficheros mediante firmas, concretamente en este caso por medio de los llamados "Magic Number".

1.1. Motivación

Existe poco software relacionado con la informática forense por lo cual este trabajo ofrece una posible contribución a la docencia y además es una oportunidad muy conveniente para ahondar en técnicas y conceptos que en la asignatura de informática forense se han visto de manera superficial.

Este proyecto es muy interesante porque incluye además de una herramienta de extracción de archivos (totalmente portable e independiente del sistema operativo) mediante búsquedas y filtros, una manera de organizar los casos, equipos y medios, así como crear grupos personalizados para buscar tipos de ficheros.

Desde el punto de vista docente cabe la posibilidad de hacer prácticas con un software en la asignatura de informática forense. Otra de las posibilidades que brinda es la de usarse como herramienta en un laboratorio para hacer análisis forenses.

Cabe remarcar también que se han tenido en cuenta algunos detalles que otros softwares forenses no suelen incluir como que se tiene en cuenta al copiar los ficheros que las rutas pueden ser demasiado largas y se da un equivalencia manteniendo la estructura física o al menos parte de ella siendo toda la información de las mismas recuperable.

1.2. Objetivos y Alcance

Uno de los objetivos de este proyecto es que la aplicación debe permitir la búsqueda de ficheros por metadatos tales como el tipo, el tamaño, las diferentes fechas (acceso,creación y modificación), propietario,texto incluido e incluso el tipo de los ficheros.

Respecto a las búsquedas por texto, será posible filtrar los resultados por medio de una cadena de caracteres alfanuméricos y se podra hacer búsquedas en cualquier fichero de texto plano en cualquier formato por ejemplo “.txt”, “.doc”, “.xls”, “.docx”, “.xlsx”, “.pdf”.

En el caso del tipo se debe llevar a cabo una identificación del tipo de fichero mediante firmas, equivalente a la herramienta Trid de Marco Pontobello, incluyendo el tipo y el porcentaje del fichero que pertenece a ese tipo (por si hay ficheros multitipados).

Por otro lado la aplicación ha de ser multiplataforma, es decir, que sea capaz de ejecutarse al menos en los principales sistemas operativos que ofrece el mercado, con el mismo ejecutable.

La aplicación también debe de copiar ficheros manteniendo la estructura original y como consecuencia la creación de un fichero, que de alguna manera muestre toda la estructura copiada para su posterior inclusión en el informe.

Modificación de los nombres largos de los ficheros cuando se copian de manera automática proporcionando una relación de equivalencia del camino original y el nuevo camino para incluirlo en un informe.

Opcionalmente se debe poder hacer una búsqueda de contraseñas en ficheros de configuración, los cuales suelen ser ficheros de texto con algún tipo palabras clave predefinidas.

La aplicación debe de llevar un componente que sea capaz de visualizar los correos electrónicos más comunes, es decir, sin tener en cuenta elementos especiales.

Capítulo 2

Marco Teórico

2.1. Informática Forense

El cómputo forense, también llamado informática forense, computación forense, análisis forense digital o examinación forense digital es la aplicación de técnicas científicas y analíticas especializadas a infraestructura tecnológica que permiten identificar, preservar, analizar y presentar datos que sean válidos dentro de un proceso legal.

La Informática forense sobre todo trata de enseñar un procedimiento a nivel científico para que las pruebas encontradas tengan validez a la hora de presentarlas ante un jurado que normalmente no está familiarizado con estas tecnologías por lo cual deben estar llevar una documentación muy específica y además el proceso debe poder ser repetido por otro especialista con los mismos resultados.

Como vemos en la definición de informática forense existen dos partes igualmente importantes, una de ellas es la recolección de pruebas en los soportes de almacenamiento masivo de datos, la otra es la generación de informes para presentarlos en un proceso legal, así que nuestro proyecto ha de centrarse por un lado en la recolección de pruebas en un medio adquirido (búsqueda de ficheros, identificación de los mismos, comprobación de fechas...) y además tendrá tambien que ser capaz de generar automáticamente los informes pertinentes, para qué con la documentación generada automáticamente por el mismo, se pueda repetir el proceso seguido con exactitud por otro experto, consiguiendo los mismos resultados, ya que solo así la prueba será válida en el proceso legal.

El proceso de análisis forense es una parte importante de la informática forense por lo que se da una pequeña explicación a continucación:

- **Identificación**
  Es muy importante conocer los antecedentes a la investigación, situación actual y el proceso que se quiere seguir para poder tomar la mejor decisión con respecto a las búsquedas y las estrategias.

Incluye muchas veces la identificación del bien informático, su uso dentro de la red, el inicio de la cadena de custodia, la revisión del entorno legal que protege el bien y del apoyo para la toma de decisión con respecto al siguiente paso una vez revisados los resultados.

- **Preservación**
  Este paso incluye la revisión y generación de las imágenes forenses de la evidencia para poder realizar el análisis.
  Al realizar una imagen forense, nos referimos al proceso que se requiere para generar una copia "bit-a-bit" (copia binaria) de todo el disco duro, el cual permitirá recuperar en el siguiente paso, toda la información contenida y borrada del disco duro.
  Para evitar la contaminación del disco duro, normalmente se ocupan bloqueadores de escritura de hardware, los cuales evitan la escritura en el disco, lo que provocaría una alteración no deseada en los medios.

- **Análisis**
  Proceso de aplicar técnicas científicas y analíticas a los medios duplicados por medio del proceso forense para poder encontrar pruebas de ciertas conductas. Se pueden realizar búsquedas de cadenas de caracteres, acciones específicas de los usuarios de la máquina, búsqueda de archivos específicos, recuperación e identificación de correos electrónicos, recuperación de los últimos sitios visitados, recuperación de la caché del navegador de Internet, etc.

- **Presentación**
  Es el recopilar toda la información que se obtuvo a partir del análisis para realizar el reporte y la presentación a los abogados, jueces o instancias que soliciten este informe.
  Es importante que haya dos tipos de informes, el técnico con el procedimiento para que otro experto pueda repetir el proceso con los mismos resultados y otro informe de los resultados sin muchos tecnicismos o frases difíciles de interpretar puesto que en el mayor de los casos una persona no experta evaluará estos resultados.

2.2. Metadatos

Metadatos (del griego meta, "después de, más allá de" y latín datum, "lo que se da") son datos que describen otros datos. En general, un grupo de metadatos se refiere a un grupo de datos que describen el contenido informativo de un objeto al que se denomina recurso.

Para varios campos de la informática, como la recuperación de información o la web semántica, los metadatos en etiquetas son un enfoque importante para construir un puente sobre el intervalo semántico.

Los metadatos se clasifican usando tres criterios:

- **Contenido:** Subdividir metadatos por su contenido es lo más común. Se puede separar los metadatos que describen el recurso mismo de los que describen el contenido del recurso. Es posible subdividir estos dos grupos más veces, por ejemplo para separar los metadatos

que describen el sentido del contenido de los que describen la estructura del contenido o los que describen el recurso mismo de los que describen el ciclo vital del recurso.

- **Variabilidad:** Según la variabilidad se puede distinguir metadatos mutables e inmutables. Los inmutables no cambian, no importa qué parte del recurso se vea, por ejemplo el nombre de un fichero. Los mutables difieren de parte a parte, por ejemplo el autor de una parte específica de un texto.

- **Función:** Los datos pueden ser parte de una de las tres capas de funciones: subsimbólicos, simbólicos o lógicos. Los datos subsimbólicos no contienen información sobre su significado. Los simbólicos describen datos subsimbólicos, es decir añaden sentido. Los datos lógicos describen cómo los datos simbólicos pueden ser usados para deducir conclusiones lógicas, es decir añaden comprensión.

Otra de las cosas importantes sobre los metadatos es su ciclo de vida, el cual, comprende las fases creación, manipulación y destrucción.

El análisis minucioso de cada una de las etapas saca a la luz asuntos significativos.

- **Creación:** Se pueden crear metadatos manualmente, semiautomáticamente o automáticamente. El proceso manual puede ser muy laborioso, dependiente del formato usado y del volumen deseado, hasta un grado en el que los seres humanos no puedan superarlo. Por eso, el desarrollo de utillaje semiautomático o automático facilita bastante la tarea.

  En la producción automática el software adquiere las informaciones que necesita sin ayuda externa. Aunque el desarrollo de algoritmos tan avanzados está siendo objeto de investigación actualmente, no es probable que la computadora vaya a ser capaz de extraer todos los metadatos automáticamente. En vez de ello, se considera la producción semiautomática más realista; aquí un servidor humano sostiene algoritmos autónomos con la aclaración de inseguridades o la proposición de informaciones que el software no puede extraer sin ayuda.

- **Manipulación:** Si los datos cambian, los metadatos tienen que cambiar también. Aquí se hace la pregunta: ¿quién va a adaptar los metadatos? Existen modificaciones que pueden ser manejadas de forma sencilla y automática, pero hay otras donde la intervención de un servidor humano es indispensable.

  La metaproducción, el reciclaje de partes de recursos para crear otros recursos, demanda atención particular. La fusión de los metadatos afiliados no es trivial, especialmente si se trata de información con relevancia jurídica, como por ejemplo la gestión digital de derechos.

- **Destrucción:** Además hay que investigar la destrucción de metadatos. En algunos casos es conveniente eliminar los metadatos junto con sus recursos, en otros es razonable conservar los metadatos, por ejemplo para supervisar cambios en un documento de texto.

2.3. Magic Number

Un número mágico (Magic Number) en informática se refiere a unos caracteres alfanuméricos que de manera codificada identifican un archivo, generalmente ubicados al comenzar dicho archivo o al menos en una posición fija. Su uso está extendido en entornos asociados con Unix y sus derivados, como método alternativo de identificación.

Consiste en registrar un "número mágico" dentro de un archivo para así poder identificar su tipo. En un principio, se registraban 2 bytes identificadores al principio del archivo, pero puesto que cualquier secuencia binaria no decodificada puede ser considerada como un número, cualquier característica de un formato de archivo que lo distinguiese podría ser utilizada para identificarlo.

Las imágenes GIF, por ejemplo, siempre empiezan con la representación en ASCII de GIF87a o GIF89a, dependiendo del estándar al que pertenezcan. Otros muchos formatos de archivo, en especial los de texto plano, son más difíciles de identificar por este método. Por ejemplo, los archivos HTML, pueden empezar por la cadena <html> (que no se diferencia entre mayúsculas y minúsculas), o para determinados tipos de documentos se emplea la etiqueta <!DOCTYPE, y para los XHTML, el identificador XML empieza con <?xml.

En definitiva, los archivos pueden empezar con cualquier texto aleatorio o incluso líneas vacías, pero seguirían siendo HTML.

Este enfoque ofrece mejores garantías para que el archivo se identifique correctamente, y en ocasiones puede determinar información muy precisa sobre el archivo. Aun así, es sólo útil si la interfaz empleada para acceder al archivo permite al usuario la manipulación sencilla con una variedad de métodos, como oposición al doble clic que automáticamente hace lo "correcto", ya que suele estar asociado con interfaz de línea de comandos y no con las gráficas.

Puesto que el cálculo del "número mágico" puede ser bastante complejo, y habría que analizar cada archivo una vez para cada una de los "números mágicos", este enfoque se hace bastante ineficiente, especialmente si se listan varios archivos (por el contrario, el método del nombre de archivo y métodos basados en metadatos sólo necesitan contrastar un fragmento de datos con un índice ordenado).

Y, como con el ejemplo de HTML, algunos tipos de archivos no pueden ser identificados con este método. Aun así, es el mejor método por el cual un programa puede comprobar si un archivo puede ser procesado por él de forma correcta: aún pudiendo ser, el nombre o los metadatos, alterados independientemente de su contenido, pudiendo ser comprobado con un análisis de número mágico bien diseñado que compruebe corrupciones o tipos de archivos incorrectos.

2.4. Algoritmos de Busqueda en Cadenas de caracteres

A este tipo de algoritmos también se les llama Algoritmos de patrones en un texto, algoritmos de emparejamiento de secuencias, algoritmos de casamiento de secuencias o simplemente por su nombre en inglés "string matching". Este tipo de algoritmos persiguen encontrar una o varias subcadenas en una cadena de caracteres o un texto más amplio.

2.4.1. Búsqueda por un único patrón

También llamados por su denominación en inglés "single string matching". En este tipo de algoritmos sólo se busca una subcadena a la que llamamos patrón, es decir el objetivo es encontrar todas las ocurrencias del patrón p dentro del texto. Este tipo de algoritmos se suelen agrupar en alguno de los siguientes tipos:

1. Fuerza bruta. La idea es ir deslizando el patrón sobre el texto de izquierda a derecha, comparándolo con las subcadenas del mismo tamaño que empiezan en cada carácter del texto.

2. Leer todos los caracteres del texto uno a uno modificando en cada paso algunas variables que permitan identificar posibles ocurrencias. A este tipo pertenecen los algoritmos de Knuth-Morris-Pratt, Shift-Or o búsqueda simple con autómata determinista.

3. Buscar el patrón en una ventana que se desliza a lo largo del texto. Para cada posición de esta ventana buscamos de derecha a izquierda un sufijo de la ventana que corresponda a un sufijo del patrón. A este tipo pertenecen los algoritmos de Boyer-Moore, Boyer-Moore-Horspool y Sunday Quick Search. Este tipo de algoritmos no suelen funcionar bien cuando el tamaño del patrón es pequeño y hay una probabilidad alta de encontrarlo en el texto.

4. La búsqueda se realiza de derecha a izquierda dentro de una ventana, pero en este esquema se busca el sufijo más largo en la ventana que es subcadena del patrón. Ejemplos de este tipo de algoritmos son BDM, BNDM y BOM. Este tipo de algoritmos para patrones pequeños no suelen funcionar bien.

5. Esquemas basados en funciones hash. Ejemplo de este tipo de algoritmos es el de Karp-Rabin.

2.4.2. Búsqueda por múltiples patrones

También llamados por su denominación en inglés Multiple String Matching. Ahora no tenemos un sólo patrón p a buscar sino que contamos con un conjunto $P = p_1, ..., p_r$ de patrones. La solución que se suele adoptar es la extensión de los esquemas anteriores para el caso múltiple. Por tanto tenemos los siguientes subtipos:

1. Fuerza bruta.
2. Extensión del tipo 2 de algoritmos de búsqueda simple de subcadenas. De este tipo de algoritmos son los de Aho-Corasick, Multiple Shift-And y búsqueda múltiple con autómata determinista.
3. Extensión del tipo 3 de algoritmos de búsqueda simple de subcadenas. De este tipo son los algoritmos de Commentz-Walter, Set Horspool, Wu-Manber.
4. Extensión del tipo 4 de algoritmos de búsqueda simple de subcadenas. De este tipo son los algoritmos SBOM, Multiple BNDM, DAWG-MATCH.
5. Extensión del tipo 5 de algoritmos de búsqueda simple de subcadenas.

Capítulo 3

Algoritmos de búsqueda de cadenas

A este tipo de algoritmos también se les llama Algoritmos de patrones en un texto, algoritmos de emparejamiento de secuencias, algoritmos de casamiento de secuencias o simplemente "String matching". Este tipo de algoritmos persiguen encontrar subcadenas con alguna propiedad en una cadena de caracteres que llamaremos texto.

3.1. Algoritmos de patrón unico

También llamados por su denominación en inglés "Single String Matching". En este tipo de algoritmos sólo se busca una subcadena a la que llamamos patrón, es decir el objetivo es encontrar todas las ocurrencias del patrón p dentro del texto t.

3.1.1. Shift-And

Es un algoritmo basado en prefijos y consiste en mantener un conjunto de todos los prefijos de p que coinciden con un sufijo en la lectura del texto. Los algoritmos usan paralelismo de bits para actualizar este conjunto de cada nuevo carácter de texto, este conjunto representado por el conjunto $D = d_m, ..., d_1$.

Código 3.1: Algoritmo Shift-And

```
1  Shift-And (p = p_1 p_2 ... p_m, T = t_1 t_2 ... t_n)
2  Preprocesamiento:
3     Para c ∈ Σ Hacer B[c] ← 0^m
4     Para j ∈ 1...m Hacer B[p_j] ← B[p_j] | 0^{m-j} 1 0^{j-1}
5  Búsqueda:
6     D ← 0^m
7     Para pos ∈ 1...m Hacer
8        D ← ((D << 1) | 0^{m-1} 1) & B[t_pos]
9        Si D & 10^{m-1} ≠ 0^m Entonces colocamos ocurrencia en la posición pos − m + 1
10    Fin_Para
```

3.1.2. Horspool

A diferencia del anterior, este es un algoritmo de búsqueda basado en sufijos, lo cual nos mantiene el peor caso igual que en el algoritmo anterior, sin embargo, teniendo en cuenta que muchas palabras tienen la misma raiz, el descarte se hace antes por sufijos, con estos saltos se consigue la sublinealidad del algoritmo.

Sin embargo hay un caso en el que este algoritmo no es eficiente: cuando un patrón es "pequeño" y hay muchas ocurrencias del mismo en el texto, ya que muchas veces llegariamos al peor caso.

Además la estructura usada en este algoritmo para representar el alfabeto es un mapa, que nos da una mayor rapidez en cuanto al acceso y comprobación.

Código 3.2: Algoritmo Horspool

```
1  Horspool (p = p_1 p_2 ... p_m, T = t_1 t_2 ... t_n)
2  Preprocesamiento:
3      Para c ∈ Σ Hacer d[c] ← m
4      Para j ∈ 1...m − 1 Hacer d[p_j] ← m − j
5  Búsqueda:
6      pos ← 0
7      Mientras pos ≤ n − m Hacer
8          j ← m
9          Mientras j > 0 AND t_{pos+j} = p_j Hacer j ← j − 1
10         Si j = 0 Entonces colocamos ocurrencia en la posición pos + 1
11         pos ← pos + d[t_{pos+m}]
12     Fin_Mientras
```

3.1.3. BNDM

Algoritmo de búsqueda de patrones basado en sufijos,en este caso, se busca de derecha a izquierda el sufijo más largo en una "ventana", para patrones largos y complejos, este algoritmo funciona mejor que los anteriores, reduce la memoria necesaria, y además es facil de extender para hacer búsqueda múltiple de patrones, aunque pierde mucha eficiencia para patrones cortos.

Código 3.3: Algoritmo BNDM

```
1  BNDM (p = p_1 p_2 ... p_m, T = t_1 t_2 ... t_n)
2  Preprocesamiento:
3     Para c ∈ Σ Hacer B[c] ← 0^m
4     Para j ∈ 1...m Hacer B[p_j] ← B[p_j] | 0^{j-1} 10^{m-j}
5  Búsqueda:
6     pos ← 0
7     Mientras pos ≤ n − m Hacer
8        j ← m, last ← m
9        D ← 1^m
10       Mientras D ≠ 0^m Hacer
11          D ← D & B[t_{pos+j}]
12          j ← j − 1
13          Si D & 10^{m-1} ≠ 0^m Entonces
14             Si j > 0 Entonces last ← j
15             Si_no colocamos ocurrencia en la posición pos + 1
16          Fin_Si
17          D ← D << 1
18       Fin_Mientras
19       pos ← pos + last
20    Fin_Mientras
```

3.2. Algoritmos multipatrón

También llamados por su denominación en inglés Multiple String Matching. Ahora no tenemos un sólo patron p a buscar sino que contamos con un conjunto $P = \{p^1, \ldots, p^r\}$ de patrones. La solución que se suele adoptar es la extensión de los esquemas anteriores para el caso múltiple.

3.2.1. Multiple Shift-And

Es la extensión para varios patrones del algoritmo Shift-And por lo que sigue una aproximación basada en prefijos. El esencial problema de este algoritmo es que la longitud de los prefijos, no es una constante,es decir, cada prefijo tiene una longitud diferente, como solución a este problema **Multiple Shift-And** comprueba las cadenas tomando la suma de las longitudes para hacer las diferentes comparaciones de una vez.

Código 3.4: Algoritmo Multiple Shift-And

```
1  Multiple Shift-And (P = {p^1, p^2, ..., p^r}, T = t_1 t_2 ... t_n)
2  Preprocesamiento:
3     Para c ∈ Σ Hacer B[c] ← 0^|P|
4     l ← 0
5     Para k ∈ 1...r Hacer
6        Para j ∈ 1...m_k Hacer B[p_j^k] ← B[p_j^k] | 0^(|P|-l-j) 10^(l+j-1)
7        l ← l + m_k
8     Fin_Para
9     DI ← 0^(m_r-1) 1 ... 0^(m_2-1) 10^(m_1-1) 1
10    DF ← 10^(m_r-1) ... 10^(m_2-1) 10^(m_1-1)
11 Búsqueda:
12       D ← 0^|P|
13       Para pos ∈ 1...n Hacer
14          D ← ((D << 1) | DI) & B[t_pos]
15          Si D & DF ≠ 0^|P| Entonces
16              Comprobar con cuál de los patrones coincide.
17              Colocamos ocurrencia en la posición pos + 1
18          Fin_Si
19       Fin_Para
```

3.2.2. Set Horspool

Es la extensión para la búsqueda de patrones múltiples del algoritmo basado en sufijos del algoritmo Horspool.

Como particularidad podemos decir que este algoritmo usa una estructura llamada "Trie" la cual genera un arbol para la búsqueda.

Por otra parte podemos indicar que solo es eficiente para alfabetos relativamente largos y un pequeño número de patrones.

Código 3.5: Algoritmo Set Horspool

```
1  Set Horspool (P = {p^1, p^2, ..., p^r}, T = t_1 t_2 ... t_n)
2  Preprocesamiento:
3     HO ← Trie(P^rv = (p^1)^rv, ..., (p^2)^rv)
4     Para c ∈ Σ Hacer d[c] ← lmin
5     Para j ∈ 1...r Hacer
6        Para k ∈ 1...m_j − 1 Hacer d[p_k^j] ← min(d[p_k^j], m_j − k)
7     Fin_Para
8  Búsqueda:
9     pos ← lmin
10    Mientras pos ≤ n Hacer
11       j ← 0, Current ← estado inicial de HO
12       Mientras pos − j > 0 AND δ_HO(t_{pos−j}, Current) ≠ θ Hacer
13          Si Current es un elemento terminal Entonces
14             Colocamos todas las ocurrencias(F(Current), pos)
15          Fin_Si
16          Current ← δ_HO(t_{pos−j}, Current)
17          j ← j + 1
18       Fin_Mientras
19       pos ← pos + d[t_pos]
20    Fin_Mientras
```

3.2.3. Multiple BNDM

Los prefijos se buscan de una manera similar al algotimo BNDM con la diferencia de que se buscan todos al mismo tiempo, para ello luego tenemos que hacer una limpieza en nuestro conjunto de bits para la busqueda y esto se hara con la máscara de bits CL

Código 3.6: Algoritmo Multiple BNDM

```
1  Multiple BNDM (P = {p^1, p^2, ..., p^r}, T = t_1 t_2 ... t_n)
2  Preprocesamiento:
3     Para c ∈ Σ Hacer B[c] ← 0^|P|
4     l ← 0
5     Para k ∈ 1...r Hacer
6        l ← l + lmin
7        Para j ∈ 1...lmin Hacer B[p_j^k] ← B[p_j^k] | 0^(|P|-l+j-1) 1 0^(l-j)
8     Fin_Para
9     CL ← (1^(lmin-1) 0)^r
10    CF ← (10^(lmin-1))^r
11 Búsqueda:
12    pos ← 0
13    Mientras pos ≤ n − m Hacer
14       j ← lmin, last ← lmin
15       D = 1^|P|
16       Mientras D ≠ 0^|P| Hacer
17          D ← D & B[t_(pos+j)]
18          j ← j − 1
19          Si D & DF ≠ 0^|P| Entonces
20             Si j > 0 Entonces last ← j
21             Si_no
22                Comprobar cual de los prefijos de longitud lmin coincide
23                  p^i necesita ser verificado si
24                  D & 0^(|P|-lmin×i) 1 0^(lmin-1) 0^(lmin×(i-1)) ≠ 0^|P|
25                Verificar las cadenas correspondientes con el texto
26                Informamos de la(s) ocurrencia(s) en pos + 1
27             Fin_Si
28          Fin_Si
29          D ← (D << 1) & CL
30       Fin_Mientras
31       pos ← pos + last
32    Fin_Mientras
```

3.2.4. Wu-Manber

El mal rendimiento del algoritmo "Set Horspool" es una consecuencia directa de que las longitudes de las comprobaciones disminuyen según nos vamos acercando a la cadena deseada, hasta el punto de tener que ir mirando caracter por caracter, lo cual es muy ineficiente.

El algoritmo Wu-Mamber salva este obstáculo leyendo bloques de caracteres lo que reduce la probabilidad de que cada bloque aparezca en muchas de las cadenas.

Wu-Maber soluciona esto con una función hash "$h1$" en la que indexa los bloques en una tabla llamada "SHIFT" y garantiza que la cadena es la que buscamos con otra función hash que indexa los bloques terminales en una tabla llamada "HASH".

Código 3.7: Algoritmo Wu-Manber

```
1  Wu-Manber (P = {p^1, p^2, ..., p^r}, T = t_1 t_2 ... t_n)
2  Preprocesamiento:
3     Computación de B
4     Construcción de las tablas hash SHIFT y HASH
5  Búsqueda:
6     pos ← lmin
7     Mientras pos ≤ n Hacer
8        i ← h_1(t_{pos-B+1} ... t_{pos})
9        Si SHIFT[i] = 0 Entonces
10          list ← HASH[h_2(t_{pos-B+1} ... t_{pos})]
11          Comprobar con cuál de los patrones coincide.
12          pos ← pos + 1
13       Si_no pos ← pos + SHIFT[i]
14       Fin_Si
15    Fin_Mientras
```

3.3. Comparativa de Algortimos

Para cada campo de la tabla se han hecho 10 ejecuciones y el tiempo es la media aritmetica de los tiempos "real" y "user" proporcionados por el comando "time -p" ejecutados en un sistema "Linux versión 4.8.2-1-ARCH".

Texto	Tamaño Texto (B)	Patrones
Fragmento del Quijote	4.096	Congojarte Quijote Por
Piensa en Haskell	1.396.097	Haskell Action
inserts.sql	7.444.889	Word MAGNUM

Tabla 3.1: Tabla de búsquedas: Tamaños y Patrones a buscar

Para patrones simples se coge el primero de los patrones y para patrones múltiples se hara una búsqueda sobre todos los patrones a la vez.

3.3.1. Shift-And

El coste del algoritmo Shift-And es de $\Theta(n)$, asumiendo que todas las operaciones de nuestro algoritmo pueden ejecutarse en un tiempo constante y siendo n el tamaño del texto en el que buscamos los patrones. Para cada uno de los tamaños, como son ficheros diferentes ha sido necesario hacer unas búsquedas diferentes.

Tamaño Texto (B)	Tiempo Total (s)	Tiempo Búsqueda (s)
4.096	0.13	0.014
1.396.097	0.50	0.070
7.444.889	1.30	0.794

Tabla 3.2: Comparativa de tiempos y uso de memoria respecto al tamaño del texto del algoritmo Shift-And

3.3.2. Horspool

En este algoritmo se ofrece una mejora en cuanto al anterior porque aunque el coste del peor caso es $O(3n)$, siendo n el tamaño del texto en el que buscamos, el mejor caso es sublineal, en este caso $O(m/n)$, siendo m el tamaño de nuestro patrón.

Tamaño Texto (B)	Tiempo Total (s)	Tiempo Búsqueda (s)
4.096	0.12	0.003
1.396.097	0.15	0.010
7.444.889	0.40	0.070

Tabla 3.3: Comparativa de tiempos y uso de memoria respecto al tamaño del texto del algoritmo Horspool

3.3.3. BNDM

El algoritmo BNDM tiene un coste en el peor caso de $O(m \cdot n)$ siendo m el tamaño de nuestro patrón y n el tamaño del texto en el que se hace la búsqueda, sin embargo en el mejor caso el algoritmo tiene un coste de $O(\frac{n \cdot log_{|\Sigma|} m}{m})$.

Sin embargo como podemos ver en la comparativa que el tiempo de búsqueda es muy superior al Horspool, ya que este algoritmo solo es eficiente para patrones grandes.

Tamaño Texto (B)	Tiempo Total (s)	Tiempo Búsqueda (s)
4.096	0.11	0.006
1.396.097	0.68	0.062
7.444.889	1.05	0.387

Tabla 3.4: Comparativa de tiempos y uso de memoria respecto al tamaño del texto del algoritmo BNDM

3.3.4. Multiple Shift-And

El algoritmo Multiple Shift-And conserva su coste, sin embargo la mejora reside en que este algoritmo sirve para múltiples patrones, luego seguiremos teniendo un algoritmo con un coste de $\Theta(n)$.

Tamaño Texto (B)	Tiempo Total (s)	Tiempo Búsqueda (s)
4.096	0.17	0.019
1.396.097	1.02	0.89
7.444.889	1.57	1.048

Tabla 3.5: Comparativa de tiempos y uso de memoria respecto al tamaño del texto del algoritmo Multiple Shift-And

3.3.5. Set Horspool

En esta extensión para múltiples patrones del algoritmo Horspool el coste del mismo es el siguiente $\Theta(n \cdot m)$.

Tamaño Texto (B)	Tiempo Total (s)	Tiempo Búsqueda (s)
4.096	0.12	0.006
1.396.097	0.69	0.088
7.444.889	1.81	0.258

Tabla 3.6: Comparativa de tiempos y uso de memoria respecto al tamaño del texto del algoritmo Set Horspool

3.3.6. Multiple BNDM

Con los algoritmos basados en prefijos como el "Multiple BNDM" conseguimos un buen coste en el peor caso: $O(n \cdot m \cdot \left\lceil \frac{m^2}{r} \right\rceil)$ sin embargo el peor caso es lineal en todo el texto, además es complicado de implementar y en la practica los resultados son bastante pobres.

Tamaño Texto (B)	Tiempo Total (s)	Tiempo Búsqueda (s)
4.096	0.12	0.006
1.396.097	0.62	0.120
7.444.889	1.43	0.776

Tabla 3.7: Comparativa de tiempos y uso de memoria respecto al tamaño del texto del algoritmo Multiple BNDM

3.3.7. Wu-Mamber

El algoritmo Wu-Mamber es una variación del Set Horspool sin embargo usa técnicas hash con lo cual permite un ahorro de memoría, y por lo tanto al tener el mismo coste debería ser una mejora, pero experimentalmente podemos ver que en todos los casos el Horspool hace un mejor tiempo seguramente debido a cuestiones de la manera de implementarlo.

Tamaño Texto (B)	Tiempo Total (s)	Tiempo Búsqueda (s)
4.096	0.13	0.04
1.396.097	0.28	0.1
7.444.889	0.86	0.320

Tabla 3.8: Comparativa de tiempos y uso de memoria respecto al tamaño del texto del algoritmo Wu-Mamber

3.3.8. Conclusión de la comparativa

Viendo los resultados anotados en las tablas podemos concluir que de nuestros algoritmos de patrón múltiple, el más eficiente es el Set Horspool, de manera análoga podemos concluir que de los algoritmos de patrón único el más eficiente es el Horspool y por lo tanto seán estos dos algritmos los elegidos hacer las búsquedas en la aplicación.

Capítulo 4

Planificación y Desarrollo del Proyecto

4.1. Riesgos y plan de contingencia

A continuación se enumeran y se detallan los principales riesgos que son resultado del estudio de los mismos, el impacto que conllevan, la probabilidad de producirse y en que fase del desarrollo del proyecto pueden darse.

4.1.1. Riesgos

Nombre del riesgo	**Modificación de los requisitos.**
Detalle	Debido a cambios y/o malentendidos con el tutor es posible que sea necesario cambiar el documento de requisitos.
Contexto	Se puede dar en cualquiera de las fases.
Categoría	De proceso.
Impacto	Leve en las fases iniciales y grave en las fases finales.
Consecuencias	Dependerán de la importancia del requisito y la fase en que se produzca el cambio.

Tabla 4.1: Riesgo: Modificación de los requisitos.

Nombre del riesgo	**Retraso respecto a la planificación.**
Detalle	A causa de una mala o insufuciente planificación pueden darse retrasos inesperados, en esto influye también la falta de documentación sobre porcesos anteriores similares como guía de referencia.
Contexto	Se puede dar en cualquiera de las fases.
Categoría	De proceso, de gestión.
Impacto	Grave.
Consecuencias	Retraso en la entrega de los distintos artefactos así como del producto final.

Tabla 4.2: Riesgo: Retraso respecto a la planificación.

Nombre del riesgo	**Carga excesiva.**
Detalle	Es posible que en momentos puntuales haya una carga sobreasignada que no ha sido contemplada en alguno de los recursos debida a algún retraso previo.
Contexto	Se puede dar en cualquiera de las fases.
Categoría	De proyecto.
Impacto	Leve en las fases iniciales y crítico en las fases finales.
Consecuencias	Retraso en la entrega de los distintos artefactos así como del producto final.

Tabla 4.3: Riesgo: Carga excesiva.

4.1.2. Planes de contingencia

En este punto se incluirán los planes de contingencia para los riesgos indicados en el apartado anterior, se expone el escenario en el que puede aparecer el riesgo, los puntos del desarrollo en los que se comprueba la aparición del mismo y la estrategia seguida para mermar o en el mejor de los casos evitar los efectos perjudiciales que afectan al buen desarrollo del software.

Nombre del riesgo	**Modificación de los requisitos.**
Escenario	Las consecuencias pueden variar dependendiendo de en qué fase del proyecto se produzca la modificación y de la importancia del requisito. Repercutirá en todos los artefactos que guarden relación con dicho requisito.
Punto de comprobación	Se comprobará regularmente a lo largo del proyecto, pero especialmente en las primeras fases.
Estrategia	Protección del riesgo.
Plan de Acción	Revisión del documento de elicitación de los requisitos y el modelo de casos de uso.
Monitorización	Entrevistas y encuentros con el tutor.

Tabla 4.4: Plan de contingencia: Modificación de los requisitos.

Nombre del riesgo	**Retraso respecto a la planificación**
Escenario	Se deberá reducir en la medida de lo posible, teniendo en cuenta los recursos disponibles, el lapso de tiempo entre la planificación y el desarrollo real del proyecto.
Punto de comprobación	Se comprobará regularmente a lo largo del proyecto.
Estrategia	Reducción del riesgo.
Plan de Acción	En cada hito incluir los retrasos y documentarlos para evitarlo en casos posteriores y hacer una reasignación de recursos para mitigar el retraso.
Monitorización	En cada uno de los hitos del proyecto.

Tabla 4.5: Plan de contingencia: Retraso respecto a la planificación.

Nombre del riesgo	**Carga excesiva.**
Escenario	Es posible que en ciertos momentos puntuales debido a retrasos anteriores sea sobreasignada en ciertos recursos.
Punto de comprobación	Se comprobará regularmente a lo largo del proyecto.
Estrategia	Reducción del riesgo.
Plan de Acción	Comprobación al final de cada hito de los recursos disponibles y sus tareas asignadas para que no haya una carga excesiva sobre los mismos.
Monitorización	Entrevistas y encuentros con el tutor.

Tabla 4.6: Plan de contingencia: Carga excesiva.

4.2. Planificación del proyecto

La Planificación de un proyecto de este tipo es diferente según el caso particular y las condiciones específicas del mismo con lo cual ha de ser flexible, puesto que hay algunos riesgos para los cuales no es rentable hacer un complejo plan de contingencia para ellos.

4.2.1. Fases del Proyecto

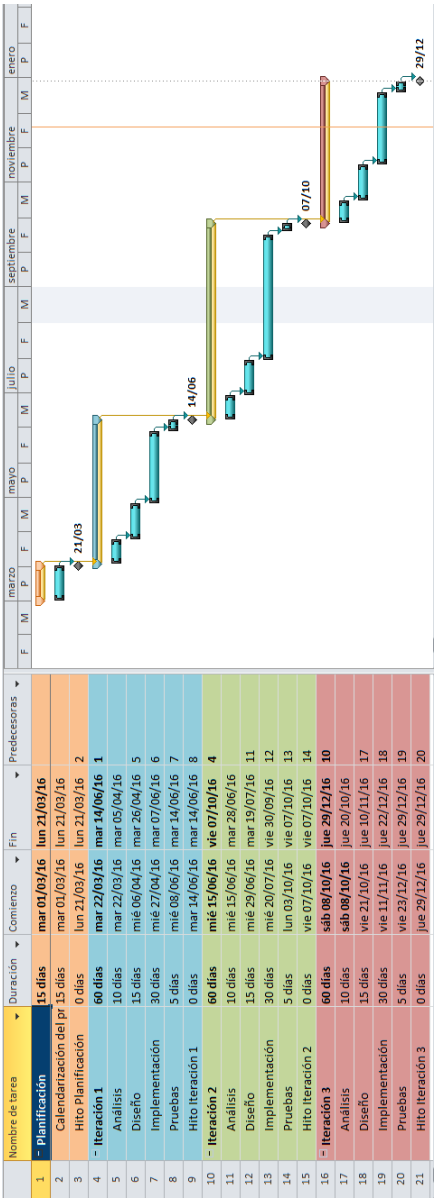
La Planificación está programada de acuerdo a 3 incrementos, se ha decidido así para minimizar los retrasos por los cambios de requisitos y de los diseños teniendo así para cada incremento todas las fases desde el "análisis" hasta las "pruebas".

Incremento	**Duración (días)**	**Fases**
Planificación	15	Planificación y gestión de riesgos
Incremento 1	60	Análisis Diseño Implementación Pruebas
Incremento 2	60	Análisis Diseño Implementación Pruebas
Incremento 3	60	Análisis Diseño Implementación Pruebas

Tabla 4.7: Fases del proyecto.

A continuación se muestra el diagrama de Gantt en el cual hemos de apuntar dos cosas, en primera instancia los fines de semana están fuera de nuestro calendario, así como el mes de agosto entero.

Lo segundo es que no se han incluido los recursos puesto que el proyecto lo ha llevado a cabo una sola persona y por lo tanto en cada uno de los puntos los recursos para esa persona alcanzarán el 100 % y no habrá recursos sobreasignados.



4.2.2. Actividades del Proyecto

Vamos a indicar las actividades de uno de los incrementos ya que todos seguirán el mismo patrón y de la planificación. Las tareas que componen la planificación son las siguientes:

- Gestión de las fases del proyecto
- Detección de riesgos
- Compilación del planes de contingencia
- Identificación de las tareas pertinentes
- Calendarización del proyecto

Por otro lado tenemos los incrementos y las tareas que conforman cada una de las fases de la misma:

- Análisis
  - Identificación de los "actores"
  - Elicitación y documentación de requisitos
  - Compliación o modificiación del los casos de uso
  - Compilación o modificiación del modelo del dominio
  - Compilación o modificiación de los modelos dinámicos
  - Compilación o modificiación de los modelos entidad-relación
- Diseño
  - Compliación o modificiación del modelo arquitectonico
  - Identificación de patrones de diseño
  - Compliación o modificiación del modelo de dominio detallado
  - Compliación o modificiación del modelo relacional
- Implementación
  - Codificación del diseño
  - documentación del código
  - Generación de la nueva versión para las pruebas
- Pruebas
  - Documentación de la batería de pruebas
  - Anotación de fallos para el siguiente incremento

4.2.3. Presupuestos

El presupuesto se calculará según el método COCOMO, para lo cual primero estimaremos el número aproximado de lineas de codigo que tendrá nuestra aplicación.

Entradas: 2	**Salidas: 2**	**Consultas: 7**	**Fich. Internos: 0**	**Fich. Externos: 1**
Archivos	Archivos	Búsquedas/ log		Base de datos
Teclado	Mensajes	Equiv. Copia/ log		
		Casos/ Base de datos		
		Grupos Base de datos		

Tabla 4.8: Cálculo de los Puntos de Función no Ajustados (PFNA)

	Bajo	**Medio**	**Alto**	**Total**
Entradas	2 (x3)	0 (x4)	0 (x6)	6
Salidas	2 (x4)	0 (x5)	0 (x7)	8
Consultas	7 (x3)	0 (x4)	0 (x6)	21
Fich. Internos	0 (x7)	0 (x10)	0 (x15)	0
Fich. Externos	0 (x5)	0 (x7)	1 (x10)	10
			Total	**45**

Tabla 4.9: Cálculo de los Puntos de Función no Ajustados II (PFNA)

Ajuste de los PFNA mediante un factor de ajuste, con un valor entre 0–5, calculado sobre 14 características generales de los sistemas.

Factor de ajuste	**0 – 5**	**Complejidad**	**0 – 5**
Comunicación de datos	0	Funciones distribuidas	0
Rendimiento	2	Gran carga de trabajo	3
Frecuencia de transacciones	3	Entrada on-line de datos	0
Requisitos de manejo del usuario final	2	Actualizaciones on-line	0
Procesos complejos	1	Utilización de otros sistemas	0
Facilidad de mantenimiento	4	Facilidad de operación	4
Instalación en múltiples lugares	5	Facilidad de cambio	4
		Total	**28**

Tabla 4.10: Ajuste de puntos de función

$FA = 0,45 + (0,01 \times 28) = 0,45 + 0,28 = 0,73$
$PF = FA \cdot PNFA = 0,73 \cdot 45 = 32,85$

Cada punto de función equivale aproximadamente a 53 líneas de código en Java. Por lo que el número total de líneas de código será: $53 \cdot 32,85 = 1.741\ LDC = 1,74\ KLCD$

Emplearemos como entorno de desarrollo típico el modelo rígido por lo tanto con esta información, obtenemos un esfuerzo nominal de:
$PM = 2,8 \cdot 1,74^{(1,2)} = 5,44 \approx 6\ personas/mes$

Y un tiempo de desarrollo de:
$TD = 2,5 * 5,44^{(0,32)} = 4,2\ meses$

Para ajustar aún más estos valores de podría usar una tabla de ajustes, sin embargo no habrá ningún cambio notable y por lo tanto cogeremos estos valores.

Aplicando estos datos a los costes de personal contamos con los siguientes resultados considerando un sueldo medio de 1400 euros:
$Coste : 6\ personas/mes \cdot 1.400 € = 8.400 €$

Respecto a los costes de material, como se ha usado como entorno de desarrollo NetBeans y el sistema operativo es linux, y en ambos sus licencias son gratuitas, solo nos queda establecer el coste del equipo, el cual haremos una estimación sobre el coste real en el momento en el que fué adquirido:

Lenovo G570 Precio: 560 € Uso 20 % Coste: 110 €

Sumando todo esto llegamos al coste total de:
Coste total: $8.400 € + 110 € = 8.510 €$

Teniendo en cuenta que solo una persona hace realmente el trabajo es normal que este se retrase y que se incremente el coste, sin ambargo nos quedaremos con esta estimación puesto que se supone que en un caso real trabajarían 6 personas los 4 meses de duración.

Capítulo 5

Análisis

5.1. Actores

Puesto que la Aplicación no requiere una identificación de la persona que interactua con ella, solo tendremos un único Actor, y como está será usada para investigar casos, llamaremos al actor "Investigador":

ACT-001	**Investigador**
Autores	Juan Miguel Celorrio.
Descripción	Este actor representa a los investigadores que utilizarán el sistema para hacer un análisis detallado del caso y generar los informes necesarios para satisfacer las buenas prácticas de la Informática Forense.

Tabla 5.1: Actor: Investigador.

5.2. Requisitos

5.2.1. Requisitos Funcionales

Un requisito funcional define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Estos requisitos vienen dados por los objetivos del sistema y una serie de técnicas sociales aplicadas con los clientes de la aplicación.

Nuestros requisitos funcionales son los siguientes:

FR-001	**Creación de un nuevo caso**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	[FR-002] Creación de un nuevo Investigador.
Descripción	El [ACT-001] Investigador debe ser capaz de crear un nuevo [IRQ-001] Caso.

Tabla 5.2: Requisito Funcional: Creación de nuevo Caso

FR-002	**Creación de un nuevo Investigador**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	[FR-001] Creación de un nuevo Caso.
Descripción	El [ACT-001] Investigador debe ser capaz de crear un nuevo [IRQ-002] Investigador y asociarle al [IRQ-001] Caso correspondiente.

Tabla 5.3: Requisito Funcional: Creación de nuevo Investigador.

FR-003	**Creación de un nuevo equipo**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	[FR-001] Creación de un nuevo Caso.
Descripción	El [ACT-001] Investigador debe ser capaz de crear un nuevo [IRQ-003] Equipo y asociarle al [IRQ-001] Caso correspondiente.

Tabla 5.4: Requisito Funcional: Creación de un nuevo Equipo.

FR-004	**Creación de un nuevo Medio**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	[FR-003] Creación de un nuevo Equipo.
Descripción	El [ACT-001] Investigador debe ser capaz de crear un nuevo [IRQ-004] Medio y asociarle al [IRQ-003] Equipo correspondiente.

Tabla 5.5: Requisito Funcional: Creación de nuevo Medio.

FR-005	**Cargar Caso**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	
Descripción	El [ACT-001] Investigador debe ser capaz de cargar un [IRQ-001] Caso existente.

Tabla 5.6: Requisito Funcional: Cargar Caso.

FR-006	**Borrar Caso**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	
Descripción	El [ACT-001] Investigador debe ser capaz de borrar un [IRQ-001] Caso ya existente en el sistema.

Tabla 5.7: Requisito Funcional: Borrar Caso

FR-007	**Crear Grupo Personalizado**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	
Descripción	El [ACT-001] Investigador debe ser capaz de crear un nuevo [IRQ-005] Grupo Personalizado.

Tabla 5.8: Requisito Funcional: Crear Grupo Personalizado

FR-008	**Borrar Grupo Personalizado**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	
Descripción	El [ACT-001] Investigador debe ser capaz de borrar un [IRQ-005] Grupo Personalizado ya existente.

Tabla 5.9: Requisito Funcional: Borrar Grupo Personalizado

FR-009	**Modificar Caso**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	
Descripción	El [ACT-001] Investigador debe ser capaz de modificar los datos de un [IRQ-001] Caso ya existente.

Tabla 5.10: Requisito Funcional: Modificar Caso

FR-010	**Quitar Investigador**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	
Descripción	El [ACT-001] Investigador debe ser capaz de desligar un [IRQ-002] Investigador del [IRQ-001] Caso al que está asociado.

Tabla 5.11: Requisito Funcional: Quitar Investigador

FR-011	**Busqueda de ficheros por metadatos**
Autores	Juan Miguel Celorrio.
Version	1.0
Dependencias	
Descripción	El [ACT-001] Investigador debe de ser capaz de realizar una búsqueda de ficheros por sus diferentes metadatos: fecha de acceso,creación y modificación, propietario, texto contenido en el fichero, tipo de fichero,tamaño.

Tabla 5.12: Requisito Funcional: Busqueda de ficheros por metadatos

5.2.2. Requisitos No Funcionales

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos.

Los requisitos no funcionales que se han elaborado a partir de los objetivos del sistema son los siguientes:

NFR-001	**Portabilidad**
Autores	Juan Miguel Celorrio.
Version	1.0
Descripción	El sistema deberá poder ejecutarse en cualquier equipo independientemente del Sistema Operativo del mismo.

Tabla 5.13: Requisito No Funcional: Portabilidad

NFR-002	**Dependencia**
Autores	Juan Miguel Celorrio.
Version	1.0
Descripción	El sistema deberá tener el menor número de dependencias con otras aplicaciones externas.

Tabla 5.14: Requisito No Funcional: Dependencia

NFR-003	**Usabilidad**
Autores	Juan Miguel Celorrio.
Version	1.0
Descripción	El 80 % de los usuarios deben ser capaces de utilizar el sistema con una tasa de errores de un 5 %.

Tabla 5.15: Requisito No Funcional: Usabilidad

NFR-004	**Aprendizaje**
Autores	Juan Miguel Celorrio.
Version	1.0
Descripción	El tiempo de aprendizaje por un usuario debe ser menor a 2 horas.

Tabla 5.16: Requisito No Funcional: Aprendizaje

NFR-005	Rapidez de acceso
Autores	Juan Miguel Celorrio.
Version	1.0
Descripción	El tiempo de acceso para cada transacción debe ser inferior a 1 segundo.

Tabla 5.17: Requisito No Funcional: Rapidez de acceso

5.2.3. Requisitos de Información

IRQ-001	Caso
Descripción	El sistema deberá almacenar la información correspondiente al Caso.
Datos específicos	• Id. del Caso. • Nombre del Caso. • Descripción del Caso. • Lista de Investigadores. • Lista de Equipos.
Tiempo de vida	Indefinido

Tabla 5.18: Requisito de Información: Caso

IRQ-002	Investigador
Descripción	El sistema deberá almacenar la información correspondiente al Investigador.
Datos específicos	• Id. del Investigador. • Nombre y apellidos del Investigador.
Tiempo de vida	Indefinido

Tabla 5.19: Requisito de Información: Investigador

IRQ-003	Equipo
Descripción	El sistema deberá almacenar la información correspondiente al Equipo.
Datos específicos	• Id. del Equipo. • Nombre del Equipo. • Lista de Medios.
Tiempo de vida	Indefinido

Tabla 5.20: Requisito de Información: Equipo

IRQ-004	**Medio**
Descripción	El sistema deberá almacenar la información correspondiente al Medio de almacenamiento.
Datos específicos	• Id. del Medio. • Ruta de montaje del medio.
Tiempo de vida	Indefinido

Tabla 5.21: Requisito de Información: Medio

IRQ-005	**Grupo Personalizado**
Descripción	El sistema deberá almacenar la información correspondiente a un Grupo Personalizado de tipos de datos.
Datos específicos	• Id. del Grupo. • Lista de Tipos. • Descripción del Grupo.
Tiempo de vida	Indefinido

Tabla 5.22: Requisito de Información: Grupo Personalizado

IRQ-006	**Magic Number**
Descripción	El sistema deberá almacenar la información correspondiente a un Magic Number que identifique el tipo de fichero.
Datos específicos	• Id. del Magic Number. • Tipo de fichero. • Extensión del fichero. • Lista de patrones contenidos. • Lista de Global String posibles.
Tiempo de vida	Indefinido

Tabla 5.23: Requisito de Información: Magic Number

IRQ-007	**Log**
Descripción	Informe de las operaciones que se realizan en la búsqueda con objetivo de poder repetir los pasos de las mismas obteniendo los mismos resultados.
Datos específicos	Fichero de log
Tiempo de vida	Indefinido

Tabla 5.24: Requisito de Información: Log

5.3. Casos de uso

El siguiente diagrama de casos de uso muestra la funcionalidad que proporciona la aplicación, este diagrama se ha elaborado a partir de la elicitación y el análisis de los requisitos y de los actores que interactúan con el sistema.

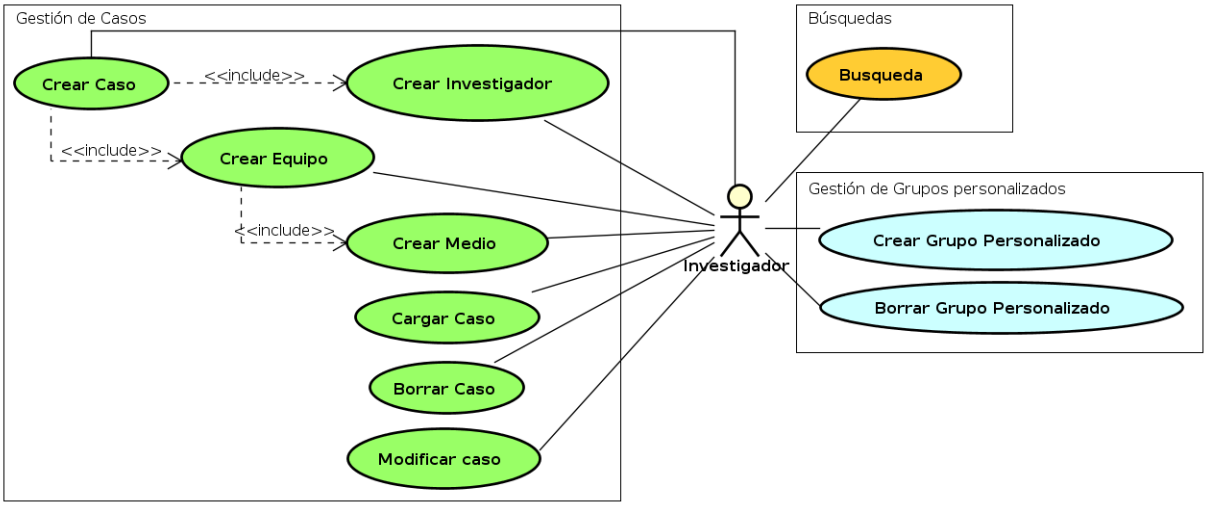


Figura 5.1: Diagrama de Casos de Uso

A continuación se muestran en detalle cada uno de los casos de uso que describen la funcionalidad proporcionada por la aplicación.

UC-001	Crear Caso
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	Ninguna
Secuencia Normal	1.- El actor introduce el nombre del caso, la descripción e introduce los investigadores involucrados. 2.- El sistema comprueba que no existe otro caso con el mismo nombre, luego crea el Caso. 3.- Para todos los investigadores de la lista realiza el [UC-002] crear investigador. 4.- Se asocia la lista de investigadores al Caso. 5.- El Actor introduce el nombre del equipo y la lista de medios. 6.- Realiza el caso de uso [UC-003] Crear equipo. 7.- Se añade la operación al informe.
Postcondición	El caso se ha creado correctamente y tendrá asociados la lista de investigadores y equipos.
Flujos Alternativos	2.1.- Si el nombre del caso está vacio, el sistema emite un mensaje de error y luego el caso de uso continúa en el paso 1. 2.2.- Si la lista de investigadores está vacia, el sistema emite un mensaje de error luego el caso de uso continúa en el paso 1.

Tabla 5.25: Caso de Uso: Crear Caso

UC-002	Crear Investigador
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	Ninguna
Secuencia Normal	1.- El actor introduce el nombre y apellidos de los Investigadores 2.- El Sistema comprueba que los datos se han introducido y son válidos. 3.- El Sistema comprueba que no existe un investigador con el mismo nombre y apellidos. 4.- El Sistema crea el investigador. 5.- Se añade la operación al informe.
Postcondición	El investigador ha creado correctamente.
Flujos Alternativos	2.1.- Si alguno de los datos no se ha introducido o no es válido, el sistema muestra un mensaje de error luego el caso de uso queda sin efecto. 3.1.- Si el investigador existe, el caso de uso queda sin efecto.

Tabla 5.26: Caso de Uso: Crear Investigador

UC-003	Crear Equipo
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	Ninguna
Secuencia Normal	1.- El Actor introduce el nombre del equipo. 2.- El Sistema comprueba que en el caso no hay ningun equipo con el mismo nombre. 3.- Si el Equipo no existe el Sistema crea el equipo. 4.- Luego el sistema realiza el caso de uso [UC-004] Crear Medios. 5.- El sistema añade la lista de medios al equipo. 6.- Se añade la operación al informe.
Postcondición	El equipo se ha creado correctamente y tendra asociado la lista de medios.
Flujos Alternativos	2.1.- Si alguno de los datos no se ha introducido o no es válido, el sistema muestra un mensaje de error luego el caso de uso queda sin efecto. 3.1.- Si el nombre del equipo existe para ese caso, el sistema muestra un mensaje de error, luego el caso de uso queda sin efecto.

Tabla 5.27: Caso de Uso: Crear Equipo

UC-004	Crear Medios
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	Ninguna
Secuencia Normal	1.- El Actor introduce a lista de medios. 2.- El Sistema comprueba que no haya dos medios iguales. 3.- Para cada medio de la lista,si el Medio no existe el Sistema crea el Medio. 4.- Se añade la operación al informe.
Postcondición	La lista de medios se ha creado correctamente y quedará almacenada en el sistema de persistencia.
Flujos Alternativos	2.1.- Si la lista de medios está vacia el sistema muestra un mensaje de error luego el caso de uso queda sin efecto. 2.2.- Si el hay dos medios iguales, el duplicado se descarta, luego el caso de caso de uso continua por el paso 2.

Tabla 5.28: Caso de Uso: Crear Medios

UC-005	Cargar Caso
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	Debe haber casos correctamente creados y almacenados en el sistema de persistencia.
Secuencia Normal	1.- El Actor selecciona la opción cargar caso. 2.- El Sistema muestra los casos disponibles. 3.- El Actor selecciona el caso que desea cargar. 4.- El Sistema muestra la información correspondiente al caso y las opciones disponibles para el mismo. 5.- Se añade la operación al informe.
Postcondición	El Caso se carga correctamente, se muestra la información del mismo y la lista de operaciones disponibles.
Flujos Alternativos	2.1.- Si no hay Casos disponibles el sistema muestra un mensaje de error y el caso de uso queda sin efecto. 2.2.- Si no se puede acceder a los Casos guardados el sistema muestra un mensaje de error y el caso de uso queda sin efecto.

Tabla 5.29: Caso de Uso: Cargar Caso

UC-006	Borrar Caso
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	Debe haber casos correctamente creados y almacenados en el sistema de persistencia.
Secuencia Normal	1.- El Actor selecciona la opción Borrar caso. 2.- El Sistema muestra los casos disponibles. 3.- El Actor selecciona el caso que desea borrar. 4.- El Sistema borra el caso de la lista y del sistema de persistencia. 5.- Se añade la operación al informe.
Postcondición	El caso quedará completamente borrado de la lista de casos y del sistema de persistencia.
Flujos Alternativos	2.1.- Si no hay Casos disponibles el sistema muestra un mensaje de error y el caso de uso queda sin efecto. 2.2.- Si no se puede acceder a los Casos guardados el sistema muestra un mensaje de error y el caso de uso queda sin efecto.

Tabla 5.30: Caso de Uso: Borrar Caso

UC-007	Crear Grupo Personalizado.
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	Ninguna.
Secuencia Normal	1.- El actor introduce el nombre del Grupo personalizado, la descripción e introduce los tipos de ficheros que desea incluir. 2.- El sistema comprueba que no existe otro Grupo personalizado con el mismo nombre, luego crea el Grupo personalizado. 3.- Para todos los tipos de ficheros de la lista realiza el [UC-00X]. 4.- Se asocia la lista de tipos de ficheros al Grupo personalizado. 5.- Se añade la operación al informe.
Postcondición	El Grupo personalizado queda creado correctamente y almacenado en el sistema de persistencia.
Flujos Alternativos	2.1.- Si alguno de los datos no se ha introducido o no es válido, el sistema muestra un mensaje de error luego el caso de uso queda sin efecto. 2.2.- Si el Grupo personalizado ya existe, el sistema muestra un mensaje de error, luego el caso de uso queda sin efecto.

Tabla 5.31: Caso de Uso: Crear Grupo Personalizado

UC-008	Borrar Grupo Personalizado.
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	Debe haber grupos personalizados correctamente creados y almacenados en el sistema de persistencia.
Secuencia Normal	1.- El Actor selecciona la opción Borrar Grupo personalizado. 2.- El Sistema muestra los Grupos personalizados disponibles. 3.- El Actor selecciona el Grupo personalizado que desea borrar. 4.- El Sistema borra el Grupo personalizado de la lista y del sistema de persistencia. 5.- Se añade la operación al informe.
Postcondición	El Grupo personalizado quedará completamente borrado de la lista de grupos personalizados y del sistema de persistencia.
Flujos Alternativos	2.1.- Si no hay Grupos personalizados el sistema muestra un mensaje de error y el caso de uso queda sin efecto. 2.2.- Si no se puede acceder a los Grupos personalizados almacenados en el sistema de persistencia, el sistema muestra un mensaje de error y el caso de uso queda sin efecto.

Tabla 5.32: Caso de Uso: Borrar Grupo Personalizado

UC-009	Modificar Caso
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	El caso debe estar correctamente cargado.
Secuencia Normal	1.- El actor selecciona la opcion modificar caso. 2.- El sistema muestra los datos actuales del caso. 3.- El actor cambia al menos el nombre del caso o la descripción y selecciona la opcion de aceptar. 4.- El sistema actualiza los datos en el sistema y en el sistema de persitencia. 5.- Se añade la operación al informe.
Postcondición	Los nuevos datos quedaran cambiados en el sistema y en el sistema de persistencia.
Flujos Alternativos	4.1.- Si no se ha cambiado ningun dato, el sistema muestra un mensaje de error y luego el caso de uso continua por el paso 2. 4.2.- Si algun dato tiene un formato incorrecto o esta vacío, el sistema muestra un mensaje de error y luego el caso de uso continua por el paso 2.

Tabla 5.33: Caso de Uso: Modificar Caso.

UC-010	Quitar investigador
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	El caso debe estar cargado correctamente.
Secuencia Normal	1.- El actor selecciona la opción Borrar Investigador. 2.- El sistema muestra los investigadores asignados al caso. 3.- El Actor selecciona varios investigadores y selecciona la opción Borrar. 4.- El sistema desvincula los investigadores del caso. 5.- Si para cada investigador no esta asociado a nigún caso, el sistema lo borra del sistema de persistencia. 6.- Se añade la operación al informe.
Postcondición	El Investigador quedará desvinculado del caso y si no está vinculado a ningún otro caso se borrará del sistema de persistencia.
Flujos Alternativos	4.1.- Si no se puede desvincular un Investigador del caso, el sistema muestra un mensaje de error, luego el caso de uso queda sin efecto. 5.1.- Si el investigador no se puede borrar del sistema de persistencia, el sistema muestra un mensaje de error, luego el caso de uso queda sin efecto.

Tabla 5.34: Caso de Uso: Quitar investigador.

UC-011	Búsqueda
Autores	Juan Miguel Celorrio.
Versión	1.0
Actor	[ACT-001] Investigador
Precondición	El caso debe estar cargado correctamente.
Secuencia Normal	1.- El Actor selecciona la opción búsqueda. 2.- El Sistema muestra los distintos tipos de búsqueda. 3.- El Actor selecciona el tipo de búsqueda deseado y los datos para el filtrado según el tipo de datos. 4.- El sistema realiza la busqueda y genera la lista de resultados. 5.- El sistema construye el informe con la lista de resultados. 6.- Se añade la operación al informe.
Postcondición	La Búsqueda se habrá generado correctamente y el sistema soportará una lista con los resultados de la misma.
Flujos Alternativos	4.1.- Si la lista de resultados está vacia, el sistema anota una mensaje de error en el informe y el caso de uso queda sin efecto.

Tabla 5.35: Caso de Uso: Búsqueda

5.4. Modelo del Dominio

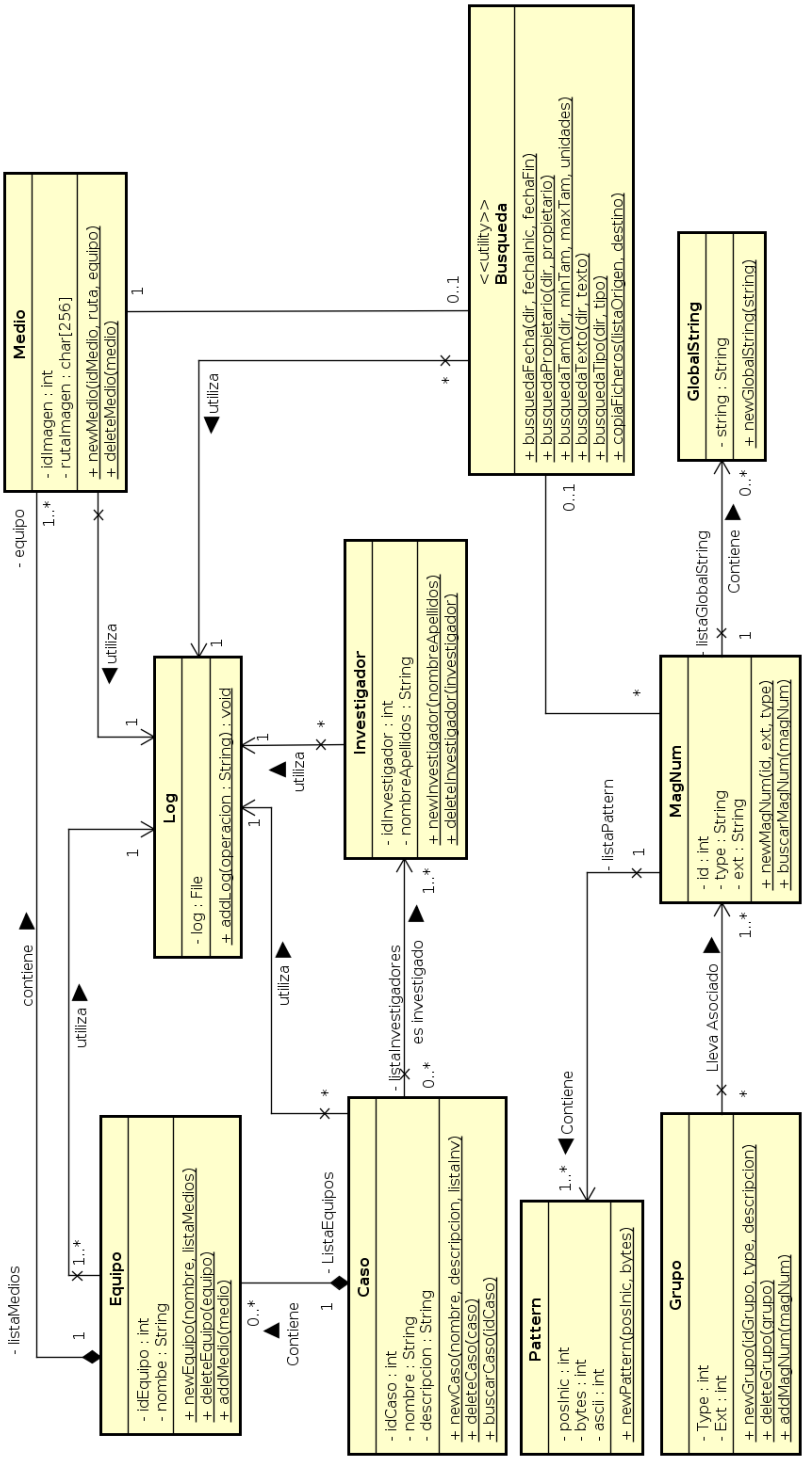


Figura 5.2: Modelo del Dominio

5.5. Modelos Dinámicos

Los modelos dinámicos se representan mediante diagramas de secuencia, los cuales permiten ver los mensajes que las diferentes clases definidas en el modelo delito dominio intercambian entre sí.

5.5.1. CU Crear Caso

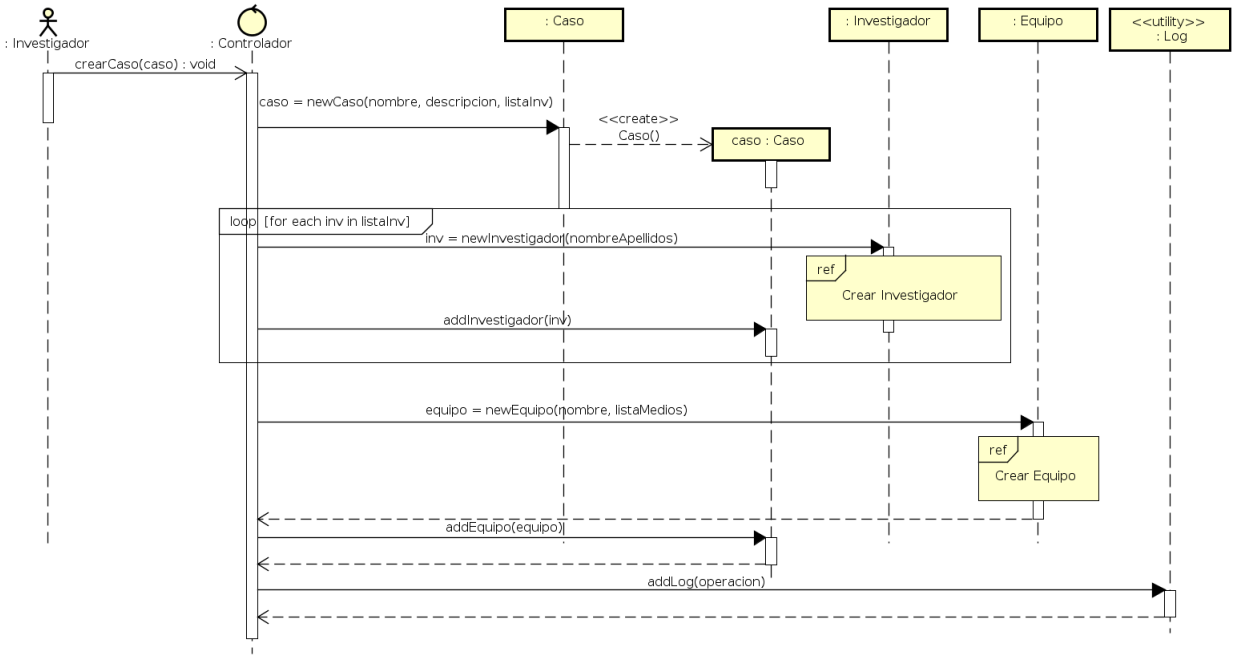


Figura 5.3: Caso de Uso: Crear Caso

5.5.2. CU Crear Investigador

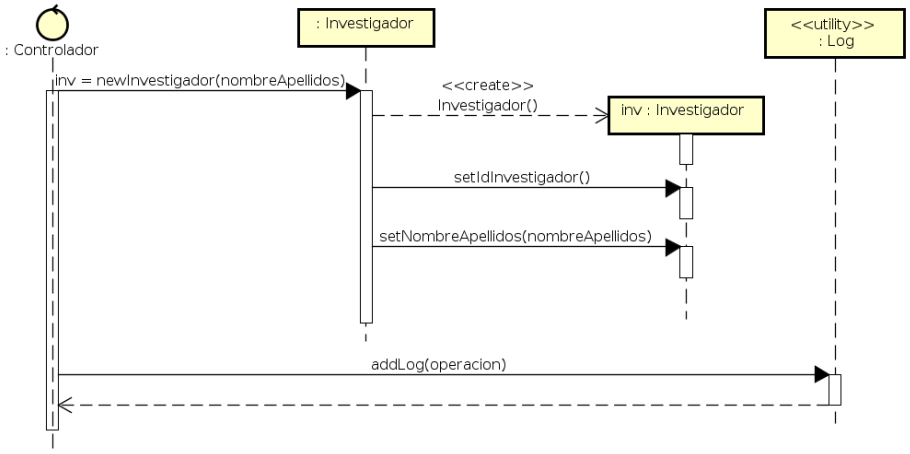


Figura 5.4: Caso de Uso: Crear Investigador

5.5.3. CU Crear Equipo

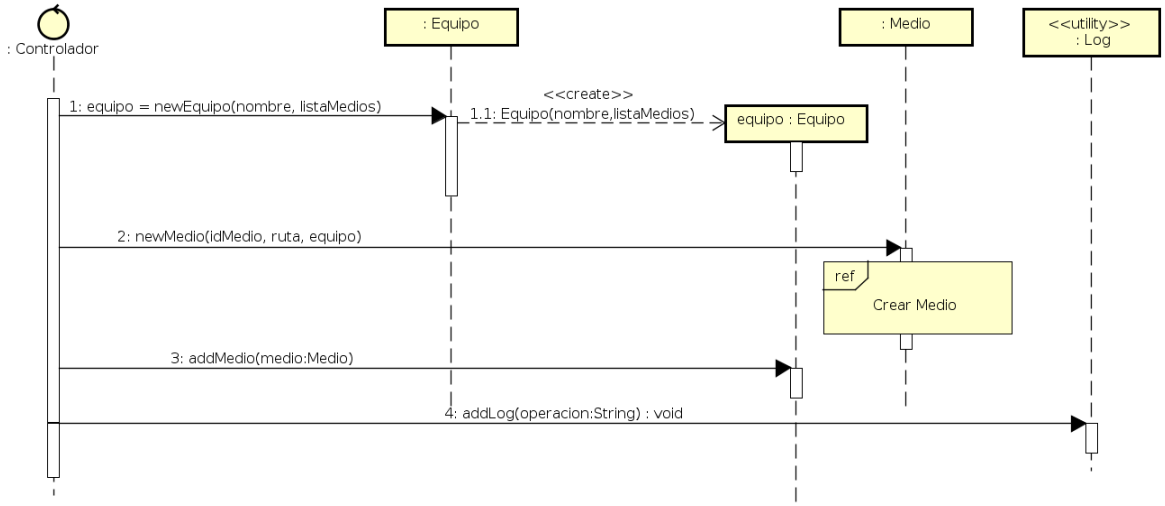


Figura 5.5: Caso de Uso: Crear Equipo

5.5.4. CU Crear Medio

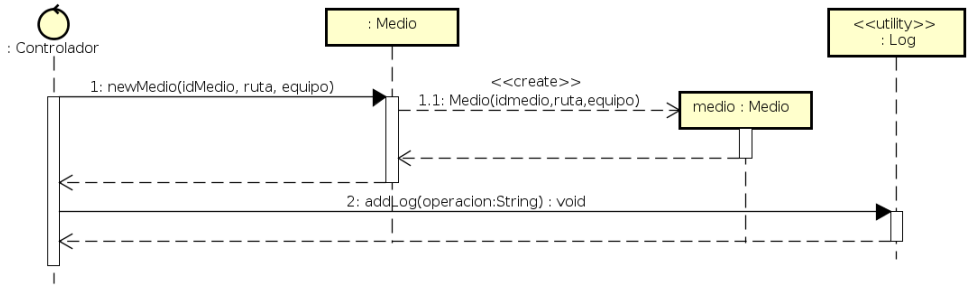


Figura 5.6: Caso de Uso: Crear Medio

5.5.5. CU Cargar Caso

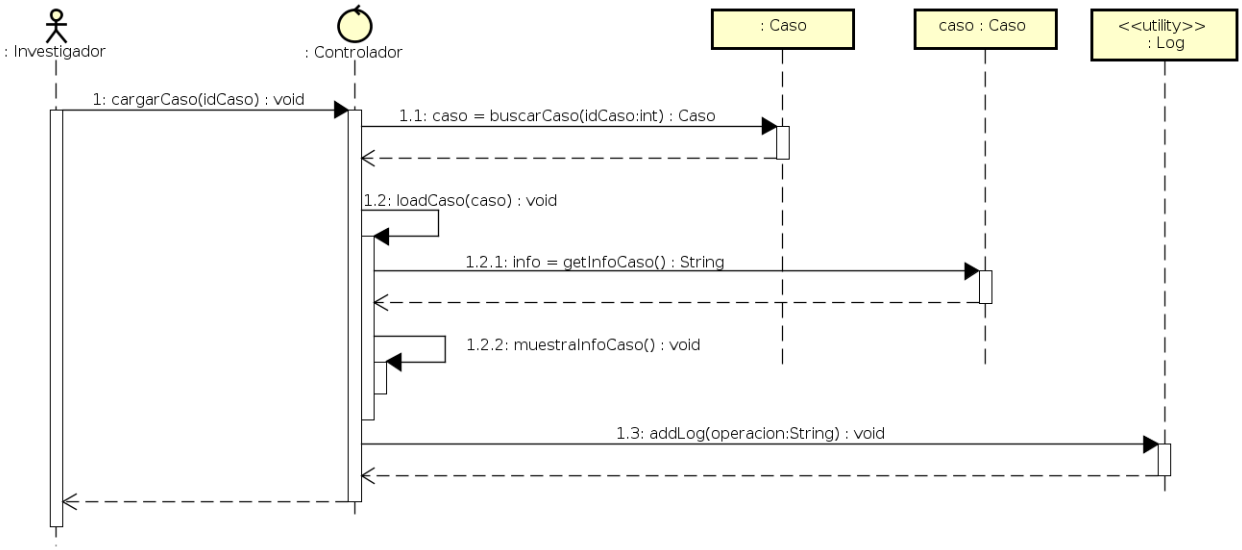


Figura 5.7: Caso de Uso: Cargar Caso

5.5.6. CU Borrar Caso

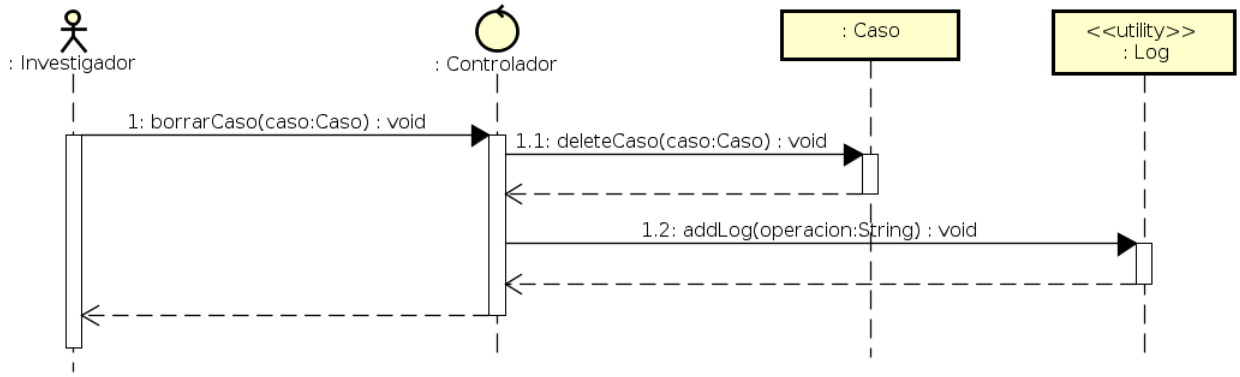


Figura 5.8: Caso de Uso: Borrar Caso

5.5.7. CU Crear Grupo

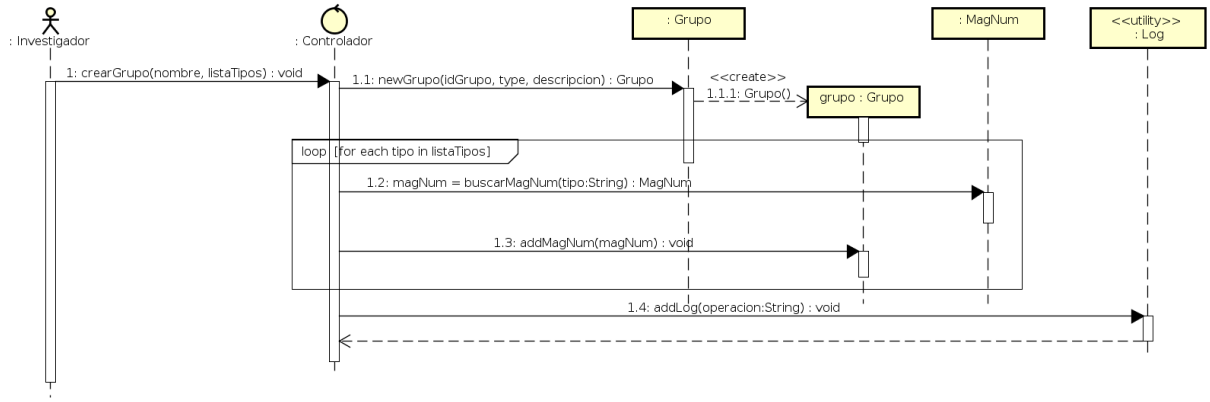


Figura 5.9: Caso de Uso: Crear Grupo Personalizado

5.5.8. CU Borrar Grupo

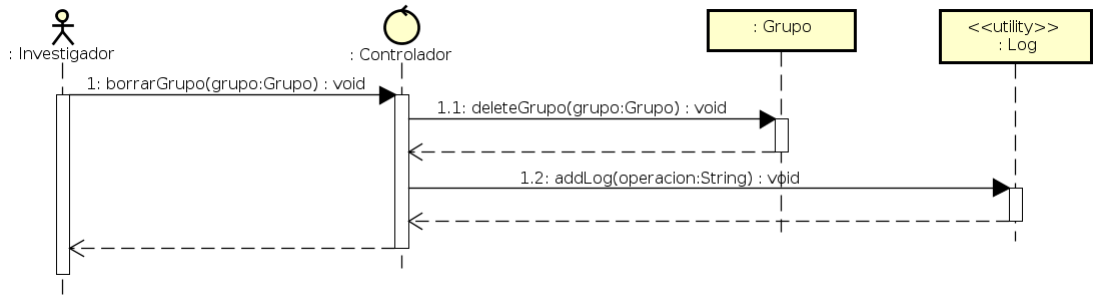


Figura 5.10: Caso de Uso: Borrar Grupo Personalizado

5.5.9. CU Busqueda

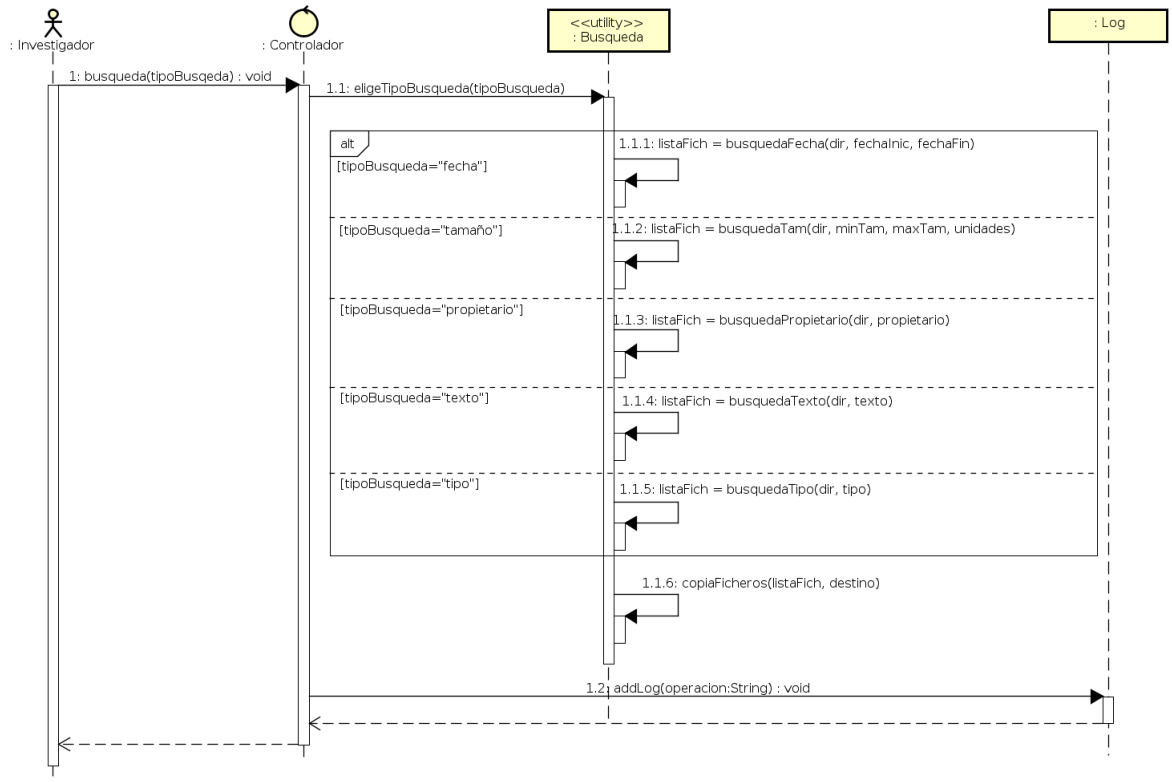


Figura 5.11: Caso de Uso: Busqueda de ficheros por metadatos

5.6. Modelo Entidad-Relación

Un modelo entidad-relación o modelo conceptual es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

Consecuentemente con eso tenemos el siguiente modelo para representar nuestro sistema de información persistente.

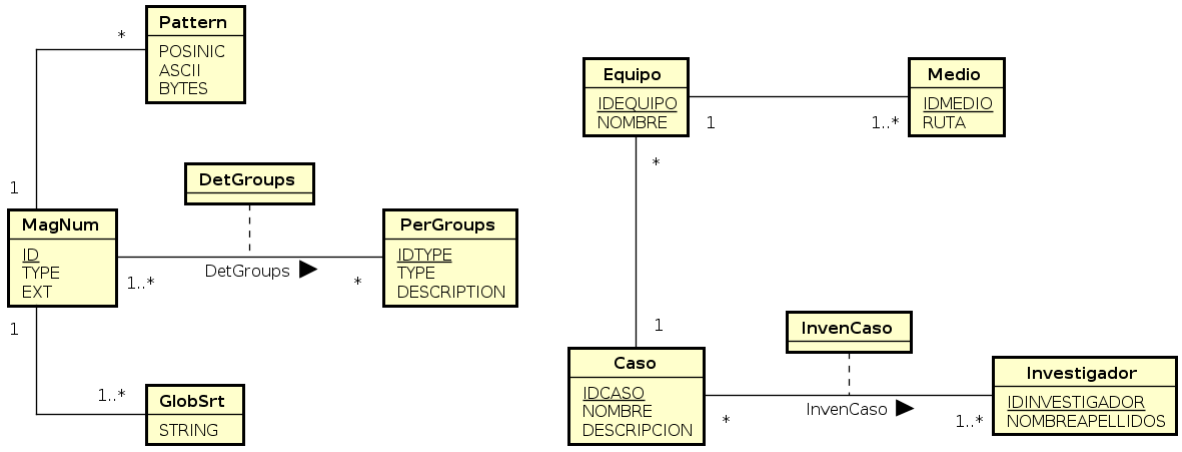


Figura 5.12: Caso de Uso: Modelo Entidad-Relación

Capítulo 6

Diseño

En esta sección del documento se explicarán en detalle la arquitectura y los diseños de las diferentes partes del sistema.

6.1. Modelo Arquitectónico

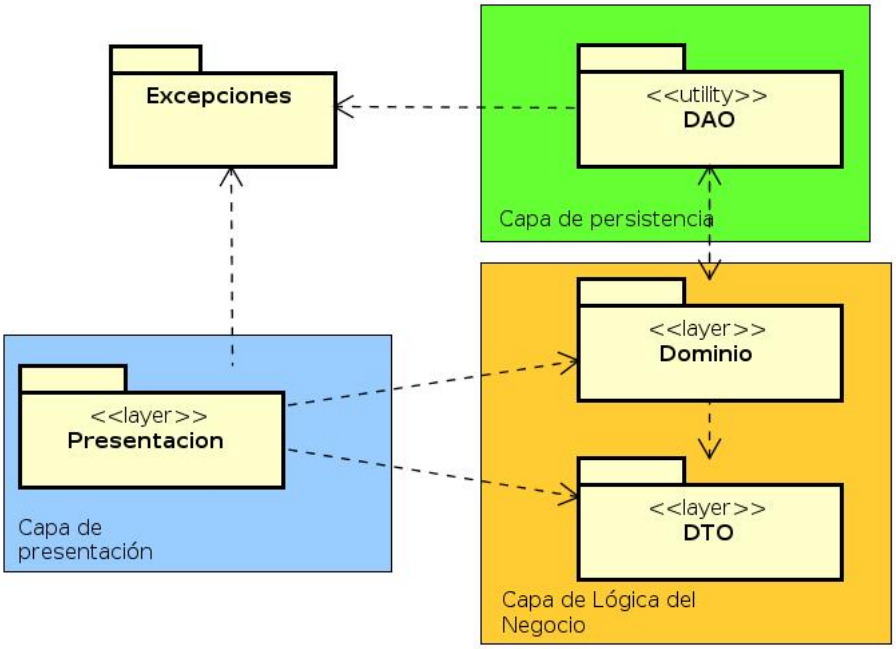


Figura 6.1: Modelo Arquitectónico

6.1.1. Capa de Presentación

En la capa de presentación mostramos los elementos correspondientes a cada una de las vistas y los gestores, que están agrupados por las diferentes funcionalidades, con lo cual tendremos un gestor relativo a los "Casos", otro gestor relativo a los "Grupos" y otro gestor relativo a las "Búsquedas".

Se han decidido hacer 3 diagramas de clases para esta capa, para poder ganar claridad y para que se vea la relación de funcionalidad de las vistas y sus correspondientes controladores.

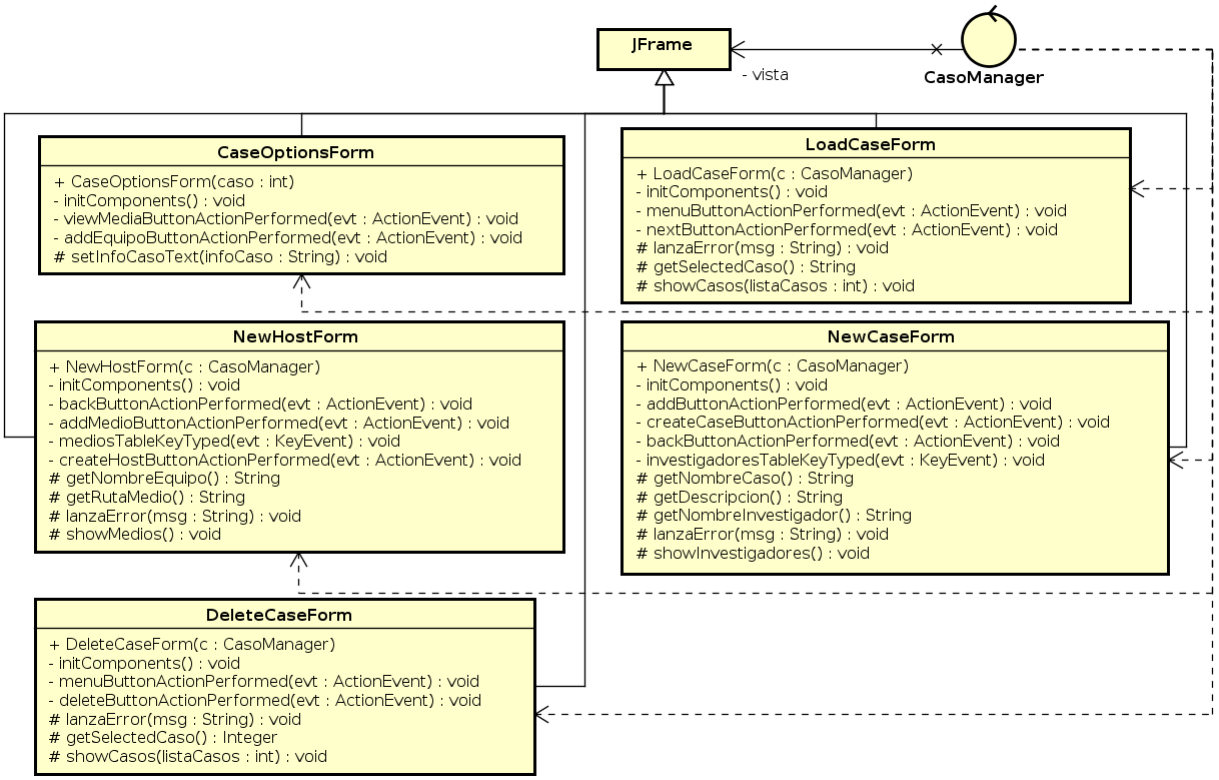


Figura 6.2: Capa de presentación I

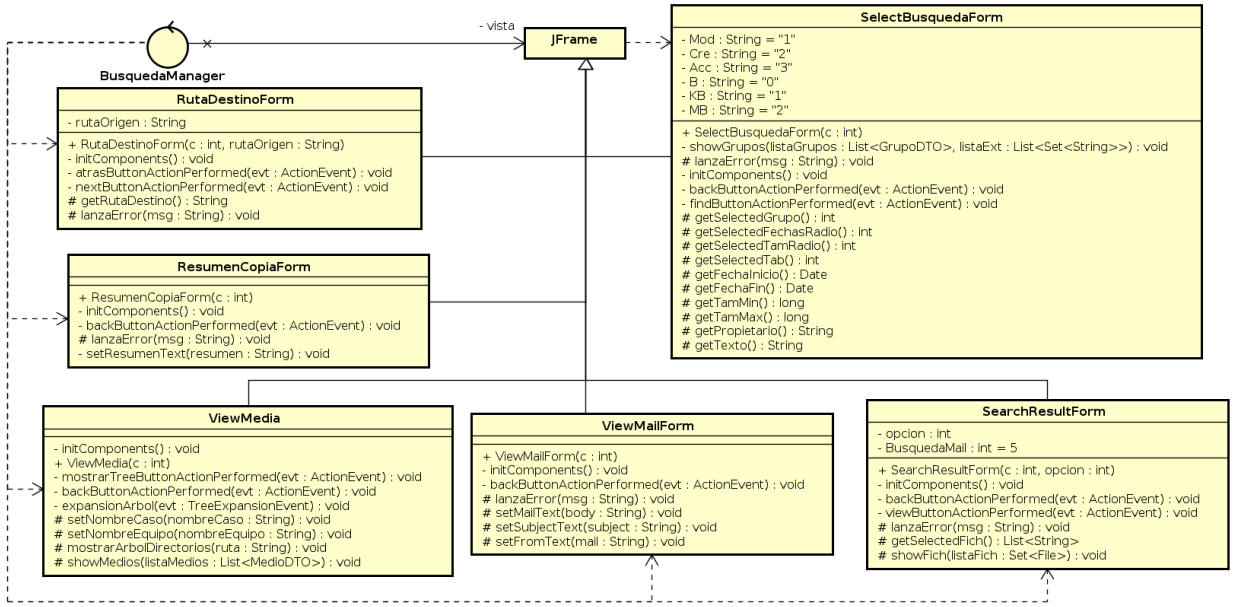


Figura 6.3: Capa de Presentación II

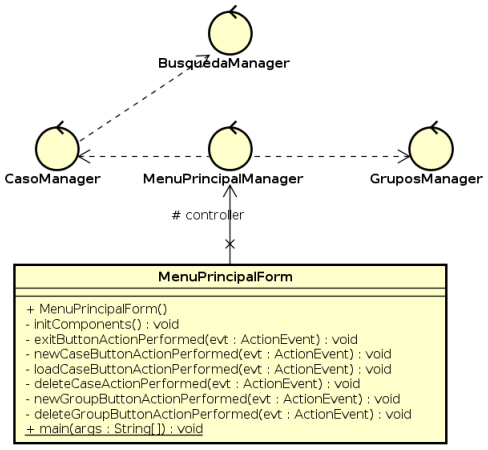


Figura 6.4: Capa de Presentación III

6.1.2. Capa de Lógica del Negocio/Dominio

En la capa de Logica del Negocio y Dominio hemos considerado incluir tanto el paquete de las operaciones estáticas sobre las clase de dominio en el paquete llamado "Dominio", como las Clases Ligeras para la transferencia de información entre capas, tambien llamados DTO en el paquete llamado de esta última manera.

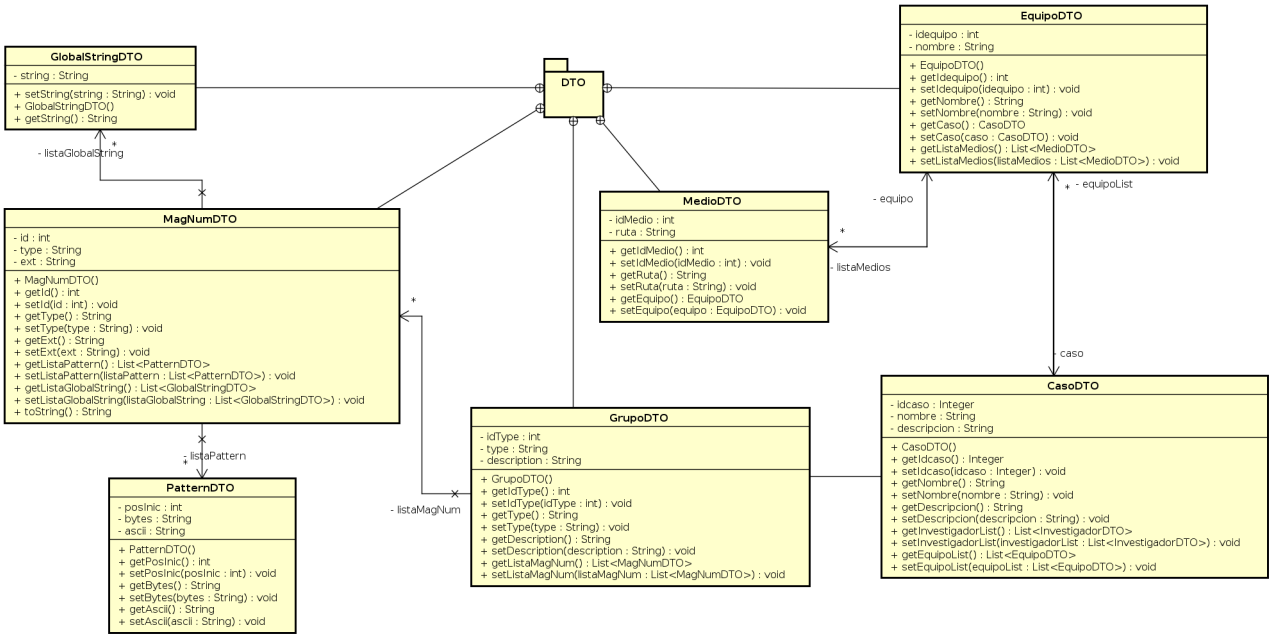


Figura 6.5: Capa DTO

6.1.3. Capa de Persistencia

En la capa de persistencia tendremos los llamados "DAOs" encargados de transformar los "DTO" en un mensaje legible para el sistema de persistencia y por medio de un "manager" o "gestor" serán capaces de almacenar estos datos de manera persistente en el sistema.

Estos "DAO" se encuentran en nuestro diseño en un paquete con ese mismo nombre.

Tendremos un "DAO" por cada uno de los DTO indicados en los diagramas de clases anteriores.

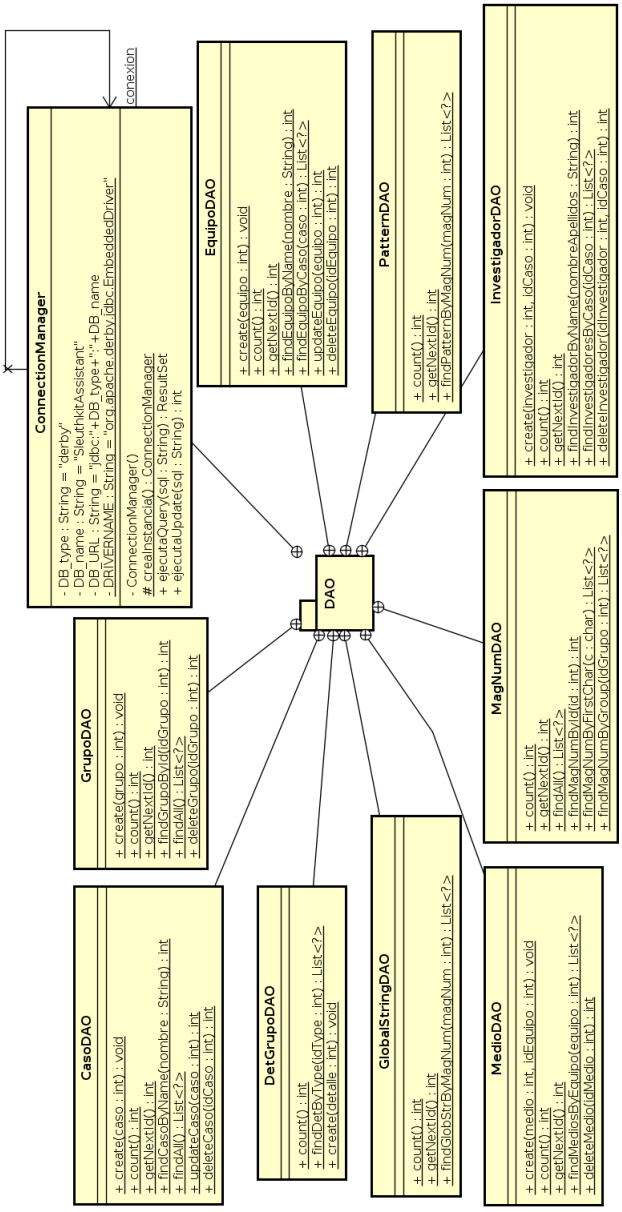


Figura 6.6: Capa DAO

6.1.4. Excepciones

El siguiente paquete no se considera una capa, sin embargo se ha considerado interesante mencionarlo ya que es una cuestión importante de diseño.

Las excepciones se han agrupado de acuerdo a la operación de la base de datos que podría fallar y por lo tanto así tener un mejor control de los flujos y por supuesto, poder personalizar los mensajes de error de una manera adecuada para que el "investigador" sea capaz de entender el

mensaje de Error con un lenguaje más natural que no requiera de unos conocimientos técnicos referentes al Lenguaje de programación.

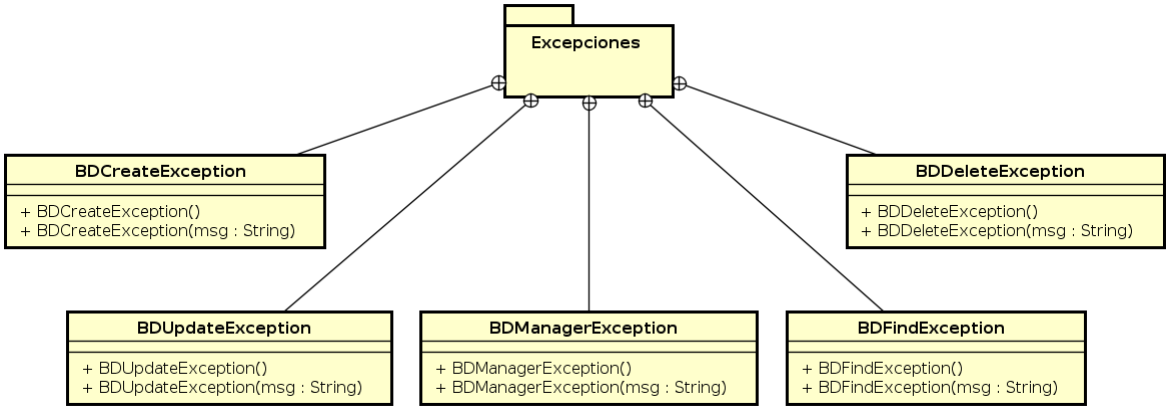


Figura 6.7: Paquete de Excepciones

6.2. Modelo Relacional

El modelo relacional es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Por consiguiente he elaborado el siguiente diseño relacional de nuestro sistema de persistencia de información basado en nuestro Modelo Entidad-Relación.

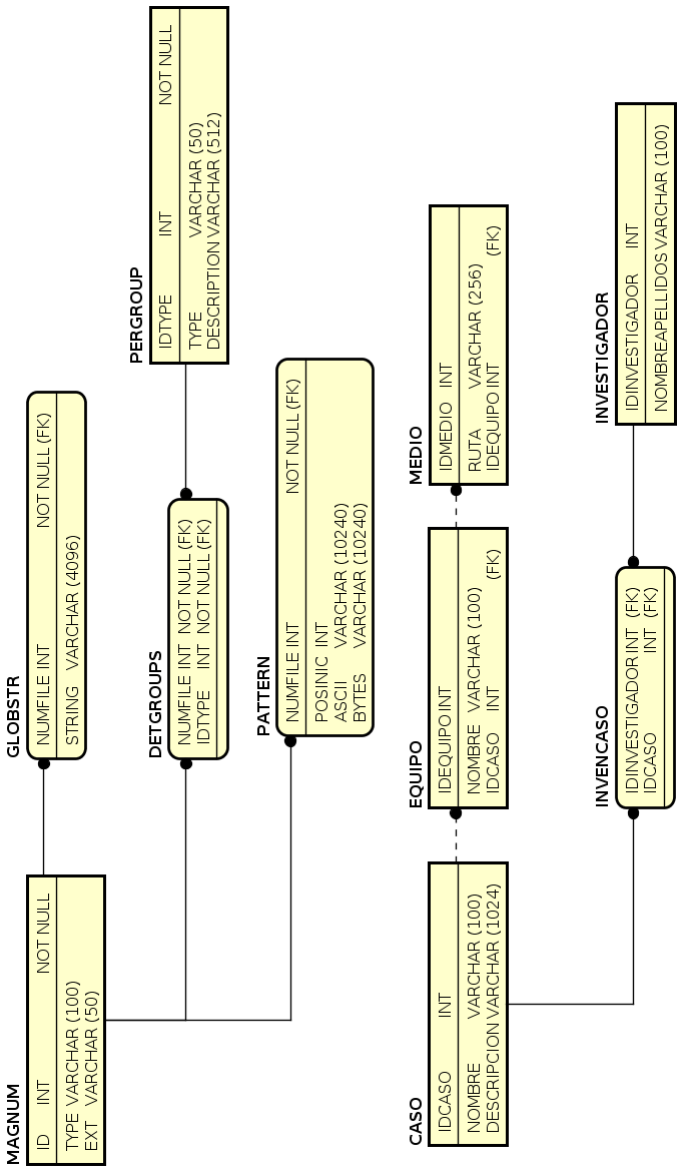


Figura 6.8: Caso de Uso: Modelo Relacional

6.3. Patrones Usados en el Diseño

6.3.1. Modelo Vista-Controlador

El **"Modelo Vista–Controlador"** (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que tienen como objetivo facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

El patrón corresponde al siguiente diagrama:

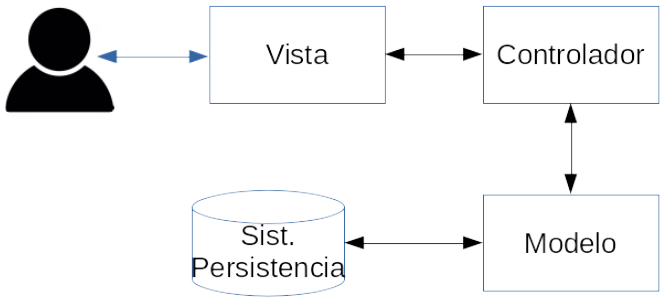


Figura 6.9: Modelo Vista-Controlador

6.3.2. Aquitectura de Capas

Es una extensión del patrón MVC en el que no solo distinguimos el modelo como algo indivisible, si no que lo consideramos divisible en diferentes capas, así la subdisión al menos sería de Presentación, Logica del Negocio o Dominio y por ultimo la capa de Persistencia que interactua con el sistema de almacenamiento de los datos, con esto conseguimos por ejemplo que un programa pueda tener diferentes interfaces y no haya que implementar un diseño diferente para cada interfaz, o si hay que migrar el sistema de almacenamiento solo tendremos que revisar la capa de persistencia, etc.

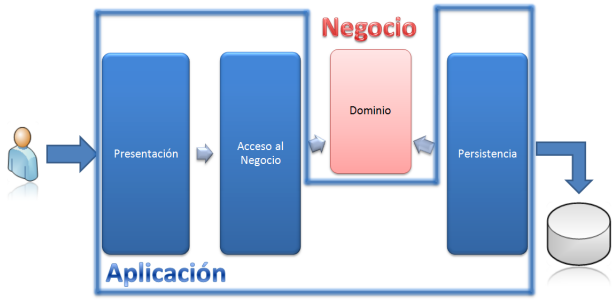


Figura 6.10: Patrones de Diseño: Patron Capas

6.3.3. Abstract Factory

El patrón "Abstract Factory" o "Factoría Abstracta" proporciona una interfaz para crear familas de objetos relacionados o que dependen entre si sin especificar sus clases concretas.

En el diseño de la aplicación se ha usado este patrón en las vistas para poder cambiar de una vista a otra en los controladores y tener una ventana activa sin necesidad de saber su clase concreta usando simplemente la clase JFrame como generalización de todas las vistas.

6.3.4. Singleton

El patrón "Singleton" garantiza que una clase tenga una instancia única y proporciona un punto de acceso global a ella.

Se usa este patrón en nuestro diseño en el gestor que comunica nuestra aplicación con el Sistema Gestor de Bases de Datos, ya que no necesita más que una instancia para ejecutar las inserciones y consultas de la base de datos, además así ahorramos memoria y controlamos de una manera eficiente los accesos a la misma.

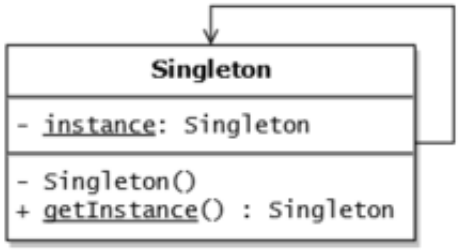


Figura 6.11: Patrones de Diseño: Singleton

6.3.5. Decorator

El patrón "Decorator" o "Decorador" asigna responsabilidades adiccionales a un objeto dinámicamente, proporcionando una alternativa felxible a la herencia para extender la funcionalidad.

Este patrón no ha sido usado explicitamente en el diseño, pero en las vistas siempre surgen elementos como las "barras de scroll" y cuadros o "frames" agrupados por capas lo cual nos lleva a pensar que este patrón se va a usar en el diseño de las vistas, aunque de una manera automatizada por el IDE.

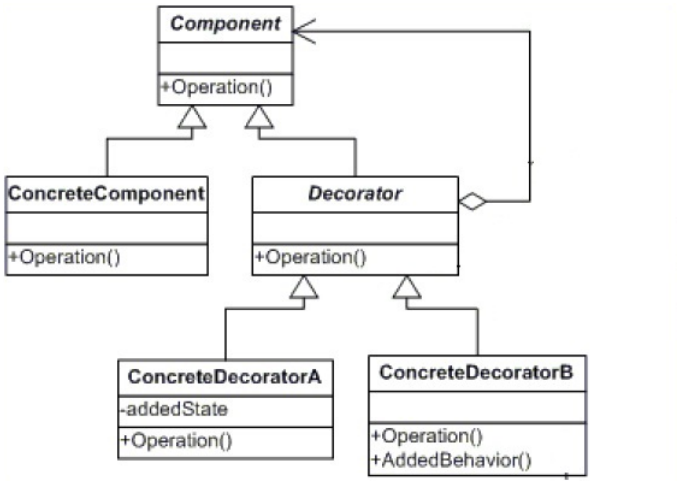


Figura 6.12: Patrones de Diseño: Decorator

6.3.6. Iterator

El patrón "Iterator" o "Iterador" proporciona un modo de acceder secuencialmente a los elementos de un objeto agregado sin exponer su representación interna.

Es otro de los patrones que no se usan explicitamente en el diseño, pero es lógico decidir usarlo para todas las "Colecciones de datos" que tenemos incluidas en los diagramas de clases en los que tenemos representadas las capas.

6.3.7. Facade

El patrón "Facade" o "Fachada" proporciona una interfaz única para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más facil de usar. En el diseño de la aplicación hacemos un uso parcial de este patrón y creamos varios controladores según la funcionalidad de la aplicación, en este caso, crearemos un "Gestor de casos", un "Gestor de búsquedas" y un "Gestor de Grupos".

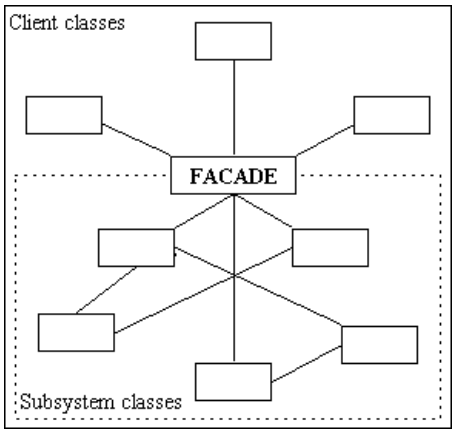


Figura 6.13: Patrones de Diseño: Facade

6.4. Reglas del Convertidor XML-SQL

Como fuente para sacar los Magic Number, se ha usado los ficheros en XML proporcionados por la pagina de "Marco Pontobello".

Como para nuestra aplicación se va a usar una base de datos relacional por motivos de integración, se ha decidido hacer un "parser" de XML a SQL muy sencillo por medio del lenguaje Lex, que nos permite con unas sencillas reglas pasar las etiquetas XML y su contenido a una sentencia SQL e incluyendole un script en bash podemos iterar este proceso para todos los ficheros XML sin necesidad de ir uno a uno.

Las siguientes reglas han sido elaboradas a partir de las expresiones regulares necesarias para captar los elementos léxicos del lenguaje XML.

Código 6.1: Reglas del Convertidor XML-SQL

```
1 TAG " <"[ a-zA-Z0-9=".]+" >"
2 CTAG " </"[ a-zA-Z0-9=".]+" >"
3 PALABRA [ a-zA-Z0-9 :!/-).(']+
```

Con estas reglas y las instrucciones correspondientes de C adicionales, se consigue almacenar el valor de las etiquetas xml, almacenarlo para tratarlo y después concatenar una o varias sentencias SQL para cada fichero, correspondiendose con nuestro esquema relacional.

6.5. ConexionBD

Este diagrama muestra como se ha implementado la conexión a la base de datos conforme se ha enseñado en la clase de Diseño de Software, se ha implementado con el patrón "Singleton" con objeto de minimizar las instancias abiertas al gestor a una instancia única y global.

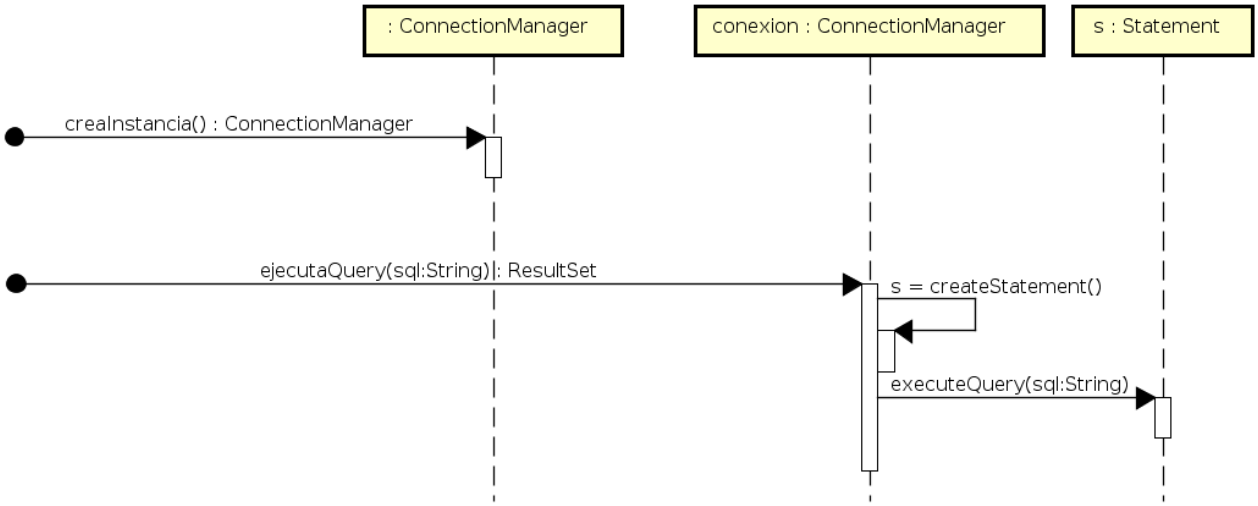


Figura 6.14: Conexion a la Base de Datos

6.6. Diseños descartados

Como primera opción se barajo la posibilidad de hacer un diseño basado en componentes con un servidor de aplicaciones para "Java EE" en concreto un "Glassfish v4", este diseño tenía muchas cosas a favor, como que era totalmente independente del Sistema Operativo, ya que se ejecutaba desde un navegador, se puede ejecutar desde un sistema remoto con una conexion al servidor, la base de datos estaría centralizada, y al estar dividido en módulos su posterior mantenimiento sería mucho más comodo, sin embargo este diseño se descartó por el motivo de que usaba demasiadas tecnologías y uno de los requisitos del sistema era el uso del menor número de tecnologías posible para su funcionamiento.

El diagrama de despliegue resultante a este diseño es el siguiente:

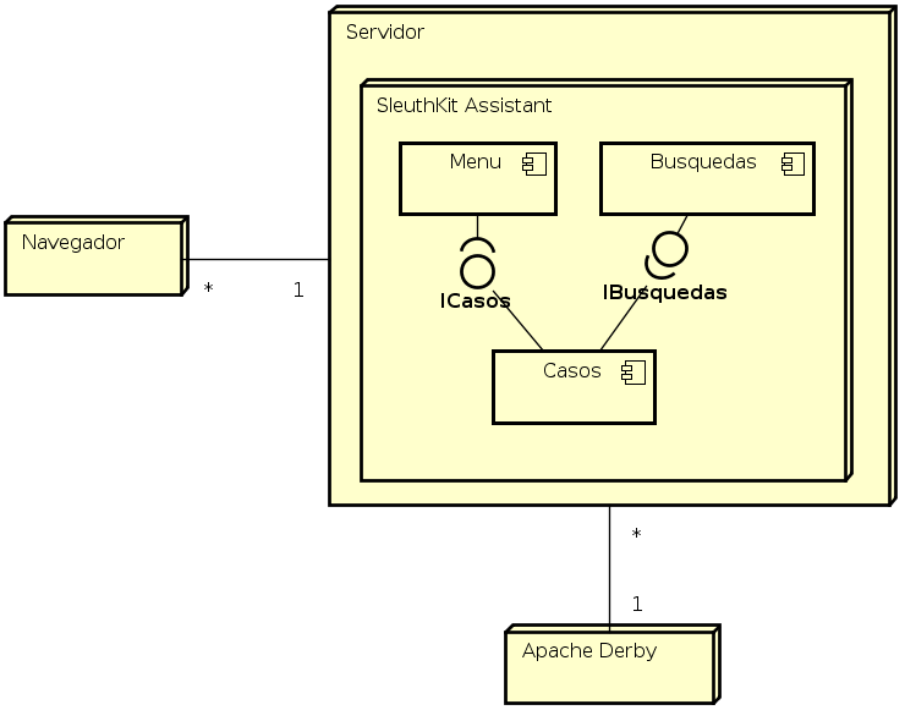


Figura 6.15: Primer diseño (descartado)

Capítulo 7

Implementación

7.1. Tecnologías requeridas

En el siguiente apartado se incluirán las tecnologías necesarias para la implemantación y el correcto funcionamiento de cada una de las partes de la aplicación.

7.1.1. Java Development Kit

Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación.

Los programas más importantes que incluye son:

- javac: Es el compilador de java, se encarga de traducir un codigo Java al lenguaje intermedio que la JVM es capaz de interpretar.
- java: Es el intérprete de java encargado de ejecutar aplicaciones compiladas por javac.
- javadoc: Es el generador de la documentación de las clases java, esta documentación estaŕa incluida con etiquetas especificas dentro del código Java.

Uno de los objetivos de este proyecto es que sea lo más portable posible para lo cual se ha decidido que lo mejor es desarrollarlo en lenguaje Java ya que podremos complilarlo en cualquier Equipo y solo llevando el ejecutable y un par de archivos más, podremos ejecutarlo en cualquier otra máquina que tenga JDK o JRE (Java Runtime Enviroment), también llamado coloquialmente JVM (Java Virtual Machine).

La versión de java usada para este proyecto es la siguiente:

```
java version "1.8.0_74"
Java(TM) SE Runtime Environment (build 1.8.0_74-b02)
Java HotSpot(TM) 64-Bit Server VM (build 25.74-b02, mixed mode)
```

7.1.2. Gestor de Bases de Datos Apache-derby

Es un módulo para la gestión de las bases de datos en aplicaciones, una de las razones por las que se ha usado es es altamente integrable con java, es muy ligero y es muy sencillo acoplar pequeñas bases de datos en aplicaciones Java.

Además nos da la ventaja de qué no necesitamos instalar aparte un Sistema Gestor de Bases de Datos aparte, ya que viene incluido en el JDK.

7.1.3. NetBeans 8.1

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un POJO que adicionalmente contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo.

Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

Además nos da la ventaja de qué no necesitamos instalar aparte un Sistema Gestor de Bases de Datos aparte, ya que viene incluido en el JDK como comentamos anteriormente.

La versión utilizada de NetBeans es la siguiente:

```
Product Version: NetBeans IDE 8.1 (Build 201510222201)
Updates: NetBeans IDE is updated to version NetBeans 8.1 Patch 1
Java: 1.8.0_66; Java HotSpot(TM) 64-Bit Server VM 25.66-b17
Runtime: Java(TM) SE Runtime Environment 1.8.0_66-b17
```

7.1.4. Java Swing

Java Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, pestañas, botones, desplegables, tablas, menús y arboles.

Se ha decidido usar esta biblioteca para implementar las interfaces de usuario de la aplicación, por su fuerte integración con Java, además se ha descartado el uso de **AWT**, que era otra posibilidad, por su menor dependecia con el Sistema Operativo, es decir, **AWT** solo tiene los componentes comunes en los diferentes sistemas operativos mientras que **Java Swing** tiene sus propios componentes, lo cual aumenta la portabilidad de la aplicación.

7.2. Pruebas

Se ha elaborado la documentación de las pruebas teniendo en cuenta dos puntos de vista diferentes:

1. Las **Pruebas de caja blanca** son aquellas realizadas sobre los procedimentos o funciones internas de un módulo de la aplicación. Se comprueba cada uno por separado para comprobar que cada una de las partes funciona correctamente y por lo tanto que la "lógica" interna de la aplicación sea correcta.

2. Las **Pruebas de caja negra** son aquellas en las que el funcionamiento interno no es transparente y por lo tanto conocemos solo la interfaz, se centran en la comprobación de la funcionalidad de la aplicación y la persona a cargo de las pruebas solo se limita a suministrar las entradas y comprobar que las salidas sean las esperadas.

7.2.1. Pruebas de caja blanca

Estas pruebas también llamadas estructurales o de caja transparente, se han ido realizando, a medida que se iba desarrollando el código, de forma que cada método y la colabración entre otras han sido probadas.

Además al haber tantos métodos no sería óptimo poner las pruebas en de caja blanca en este documento.

7.2.2. Pruebas de caja negra

Este tipo de pruebas está enfocado a comprobar que los requisitos funcionales se han cumplido. No son una alternativa a las pruebas de caja blanca, si no que son una técnica complementaria que se ha de usar por separado para detectar la mayor cantidad de fallos posibles y así poder corregirlos para un buen desarrollo de la aplicación.

Nombre del caso	Descripción	Investigador	Salida
-	-	-	Error
Bárcenas	-	-	Error
Bárcenas	-	Juan Miguel Celorrio	Exito
Bárcenas	Bárcenas fué imputado y su ordenador sobremesa ha sido incautado	-	Error
Bárcenas	Bárcenas fué imputado y su ordenador sobremesa ha sido incautado	Juan Miguel Celorrio	Éxito

Tabla 7.1: Batería de pruebas: Nuevo Caso

Nombre del Equipo	Ruta del Medio	Salida
-	-	Error
Thosiba	-	Error
Thosiba	/mnt	Éxito

Tabla 7.2: Batería de pruebas: Nuevo caso (II).

Nombre del Investigador	Salida
-	Error
Ana María Gil	Éxito
{Ana María Gil, Pablo Celorrio, Juan Miguel Celorrio}	Exito

Tabla 7.3: Batería de pruebas:Gestionar Investigadores.

Nombre del grupo	Descripción	Fich. seleccionados	Salida
-	-	$\emptyset$	Error
Documentos PDF	-	$\emptyset$	Error
Documentos PDF	PDF y similares	$\emptyset$	Error
Documentos PDF	PDF y similares	1	Éxito
Documentos PDF	-	2	Éxito

Tabla 7.4: Batería de pruebas: Crear Grupo.

Número de grupos	Grupos seleccionados	Salida
$\emptyset$	-	Error
$n >= 1$	0	Error
$n >= 1$	1	Éxito

Tabla 7.5: Batería de pruebas: Borrar Grupo.

Nombre del Equipo	Ruta del Medio	Salida
-	-	Error
HP	-	Error
HP	/	Éxito

Tabla 7.6: Batería de pruebas: Gestionar Equipos

Nombre del Caso	Descripción	Salida
-	-	Error
Bankia	-	Éxito
Bankia	Adquisición de todos los ordenadores pertenecientes al grupo Bankia	Éxito

Tabla 7.7: Batería de pruebas: Datos del Caso

Fecha Inicio	Fecha Fin	Tipo Fecha	Salida
dd/mm/aa	dd/mm/aa	Fecha de Creación Fecha modificación Fecha de último acceso	Error
-	dd/mm/aa	Fecha de Creación Fecha modificación Fecha de último acceso	Error
-	-	Fecha de Creación Fecha modificación Fecha de último acceso	Error
1/1/01	1/12/99	Fecha de Creación Fecha modificación Fecha de último acceso	Éxito
1/1/01	2/2/01	Fecha de Creación Fecha modificación Fecha de último acceso	Éxito

Tabla 7.8: Batería de Pruebas: Búsqueda por fechas

Tamaño mínimo	Tamaño máximo	Unidades	Salida
-	-	-	Error
-	-	Bytes KBytes MBytes	Error
2	1	Bytes KBytes MBytes	Éxito
1	2	Bytes KBytes MBytes	Éxito

Tabla 7.9: Batería de Pruebas: Búsqueda por Tamaño

Tipo	Salida
∅	Error
Documentos de Word	Éxito

Tabla 7.10: Batería de Pruebas: Búsqueda por Tipo

Tipo	Salida
-	Error
juacelo	Éxito

Tabla 7.11: Batería de Pruebas: Búsqueda por Propietario

Tipo	Salida
-	Error
Documentos de Word	Éxito

Tabla 7.12: Batería de Pruebas: Búsqueda por Contenido

Capítulo 8

Conclusiones

Como resultado de todo lo citado anteriormente, tenemos una aplicación de uso fácil y apariencia amigable, que a su vez nos permite hacer informes legibles a nivel humano de los pasos que vamos haciendo en la aplicación para hacer repetible el proceso por cualquier otro investigador forense, sabiendo que esto es uno de los requisitos intínsecos a las técnicas de análisis de la informática forense.

También nos da la posibilidad de hacer búsquedas e identificar ficheros de una manera sencilla a nivel de usuario y filtrar los resultados por diferentes propiedades de los metadatos y copiar los ficheros manteniendo la estructura física, también permite visualizar correos electronicos sencillos.

Como anecdota podemos apuntar que en este algoritmo en el libro de "Flexible Pattern Matching In Strings" había una errata, en el conjunto p aparecía una sola cadena, en vez de un conjunto, como este error no aparecía en la fé de erratas se envió un correo electronico al autor, el cual respondió con un agradecimiento.

Cabe destacar que toda la parte relativa a la gestión de casos, investigadores, equipos y medios no eran parte de la funcionalidad inicial de la aplicación y fueron propuestas por el alumno con el fin de aumentar la calidad, desde el punto de vista forense, de la aplicación, teniendo como referencia para esta propuesta, los apuntes de la asignatura Informática Forense.

Respecto a uso a posteriori de la aplicación será a nivel académico, por ejemplo, para preparar unas posibles prácticas de la asignatura "Informática Forense". Otro de los usos es como pequeña herramienta para comprobar si a nivel de bytes existe alguna cadena de caracteres es un fichero. Y por último se puede usar también en un laboratorio de informática forense para analizar medios que fueron hecho por medio de una Adquisición.

8.1. Objetivos Alcanzados

- Se posibilita la búsqueda de ficheros por metadatos: tipo, tamaño, fechas, propietario,...
- Permite copiar manteniendo la estructura original con la creación de un fichero "gráfico" en el que se vea la estructura copiada toda o de los ficheros/directorios marcados (para su posterior inclusión en el informe).
- Automaticamente modifica los nombres largos de los ficheros cuando se copian proporcionando una relación de equivalencia del camino original y el nuevo camino para incluirlo en un informe.
- Extrae de las características de los ficheros seleccionados a un informe: propietario, fecha de creación, modificación, tamaño...
- Permite identificar el tipo de fichero mediante firmas, equivalente a la herramienta Trid de Marco Pontobello.
- Proporciona una herramienta para buscar ficheros que contengan una cadena de caracteres alfanumérica en cualquier formato .txt, .doc, .xls,.docx,.xlsx,.pdf...
- Incluye un visor de los correos electrónicos más simples.

8.2. Conocimientos Adquiridos

He adquirido conocimientos en el tema de las firmas de ficheros como por ejemplo los Magic Number, los tipos MIME, las extensiones y como los sistemas operativos usan las mismas para saber de que tipo es cada fichero, puesto que en ninguna asignatura vemos este tema en detalle, de manera análoga también he adquirido conocimientos en el ciclo de vida de los datos y metadatos, así como el manejo de los mismos desde el punto de vista de la programación.

Además se han adquirido conocimientos sobre los distintos algoritmos de búsqueda de patrones, ya que para entenderlos tuve que repetir los ejemplos del libro a mano, también me encargué de su implementación, uso y la detección de alguna de las erratas que contenía el libro, las cuales fueron notificadas al autor por medio de un "e-mail" y que fué contestado con un agradecimiento.

De manera análoga al manejar más de un motor de base de datos para hacer la base de datos he aprendido bastante sobre el manejo de los dirver de sistemas gestores de bases de datos en Java y sobre el manejo de excepciones de los mismos, ya que cada un gestiona las excepciones de una menera diferente.

8.3. Trabajo Futuro

Una de las posibles mejoras es implementar un sistema de actualización automática de la Base de datos para añadir los nuevos tipos de ficheros incluidos en los ficheros xml, este sistema debería detectar los que ya se tienen y solo incluir los nuevos.

Por otra parte otra mejora puede ser incluir uno o varios crackeadores de contraseñas de ficheros para textos e imágenes con contraseña, que funcione en la mayor parte de las imágenes y ficheros de texto.

Respecto al visor de correos se podría mejorar de manera que que permita mostrar elementos escritos en HTML y otras tecnologías web, mostrar imagenes,etc.

Otra de las mejoras posibles es el cambio del log a una estructura parecida a CSV, para que facilmente, se pueda incluir en una hoja de calculo o en una aplicación para realizar calculos de estadísticas, muy útiles en este tipo de casos.

Teniendo el cuenta la pobre gestión que se hace de los casos cabe decir que otra mejora será incluir más información en los casos, los investigadores, los equipos y los medios.

Por último como mejora propongo implementar un sistema inteligente capaz de hacer una elección entre los distintos algoritmos de búsqueda según el número de patrones y las propiedades a buscar de los mismos.

Apéndice 1

Manual de instalación

La manera de preparar el entorno y la base de datos para la aplicación es diferente según el Sistema Operativo que se esté usando en la maquina en la que se ejecute la aplicación, sin embargo, para la ejecución nos valdrá con el fichero jar, ya que la aplicación se ejecuta en una máquina virtual de Java, lo que la hace totalmente portable.

1.1. Windows

Lo primero que debemos es asegurarnos de que tenemos instalado o bien el JRE y apache-derby por separado, o bien el JDK que ya contiene en las ultimas versiones un gestor apache-derby contenido en la carpeta %JAVAHOME %\db.

Se debe ejecutar el archivo por lotes "prepEntorno.bat" dando como parámetro la ruta de la ubicación de apache-derby.

En mi caso está incluido en el jdk: "C:\Program Files\Java\jdk1.8.0_20\db" (Las comillas solo serán necesarias si hay espacios en la ruta).

Debemos cerrar la consola y volverla a abrir para que se actualicen las variables de entorno.

Ejecutamos el script ejecutabd.bat, si no hay ningún problema este script generará la base de datos (tarda unos 5 minutos) en un directorio llamado "SleuthAssistant", este directorio debe estar siempre en la misma carpeta que el jar para que todo funcione correctamente (Por el gestor de bases de datos embebido en la applicacion).

Luego ejecutaremos el programa .jar y la aplicación debería funcionar correctamente.

1.2. Linux y macOS

Lo primero que debemos es asegurarnos de que tenemos instalado o bien el JRE y apache-derby por separado, o bien el JDK que ya contiene en las ultimas versiones un gestor apache-derby contenido en la carpeta $JAVAHOME/db.

Se debe ejecutar el archivo bash script "prepEntorno" dando como parámetro la ruta de la ubicación de apache-derby.

En mi caso está incluido en el jdk: /lib64/jvm/java-8-jdk/db

Debemos cerrar la consola y volverla a abrir para que se actualicen las variables de entorno.

Ejecutamos el script ejecutabd, si no hay ningún problema este script generará la base de datos (tarda unos 5 minutos) en un directorio llamado "SleuthAssistant", este directorio debe estar siempre en la misma carpeta que el jar para que todo funcione correctamente (Por el gestor de bases de datos embebido en la applicacion).

Luego ejecutaremos el programa .jar y la aplicación debería funcionar correctamente.

Apéndice 2

Manual de Usuario

Lo primero que encontramos cuando ejecutamos la aplicación es un menú principal en el cual tenemos dos Grupos de opciones, por un lado tenemos las opciones referentes a los casos que son: “Crear Caso”, “Cargar Caso” y “Borrar Caso”.

Y por otro lado tenemos las opciones referentes a los Grupos personalizados que son: “Crear Grupo” y “Borrar grupo”.

Adicionalmente tenemos un botón para salir de la aplicación, en cada una de las pantallas que veremos a continuación se dará la posibilidad de volver al menú principal y desde este se podrá cerrar la aplicación.

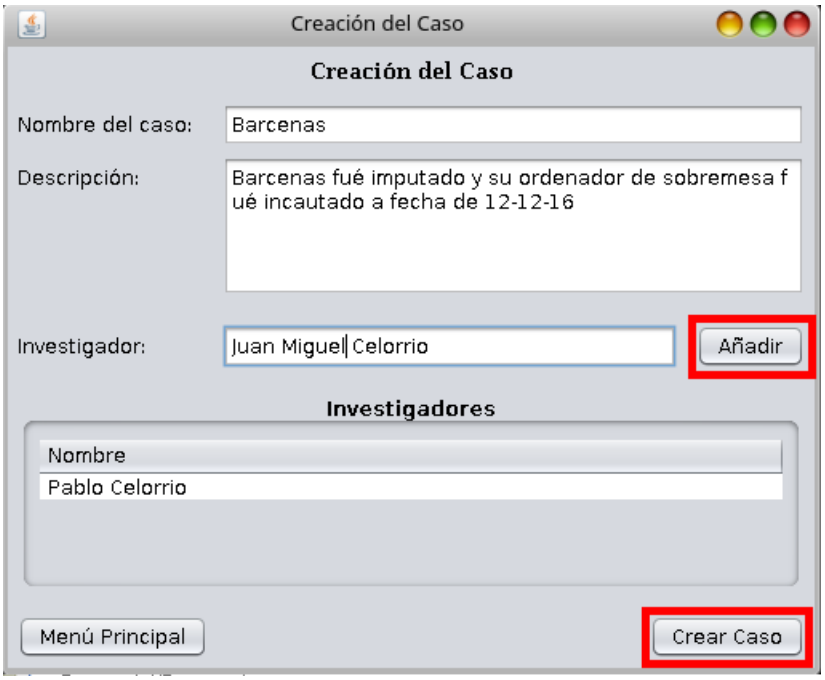
2.1. Creación de Nuevo Caso

Para ello empezaremos seleccionando la opción “Nuevo caso” ubicada en la parte dedicada a los casos en el menú principal.

Tras esto se abrirá un formulario con varios campos que debemos rellenar, al menos debemos rellenar el campo “Nombre del caso”.

El campo “Descripción” es opcional sin embargo suele ser conveniente rellenarlo y dar una descripción un poco más detallada del caso.

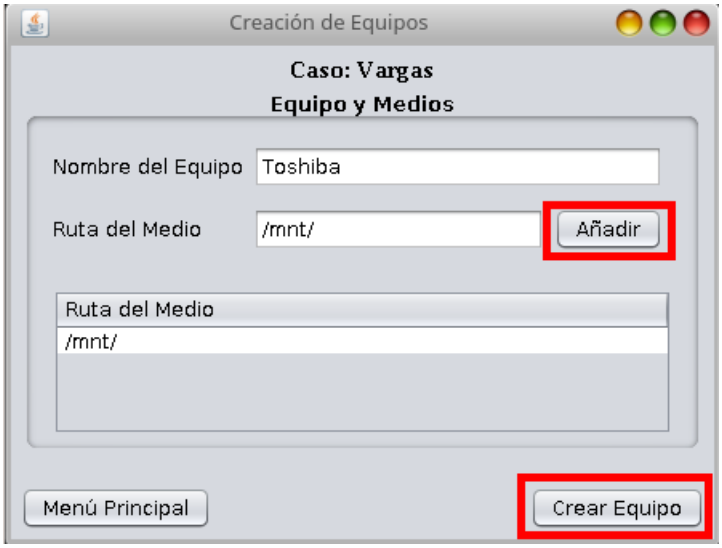
También debemos añadir Investigadores a la lista introduciendo el nombre y los apellidos del investigador, y pulsando en el botón “Añadir” (Este paso se debe repetir por cada uno de los investigadores).



Después pulse el botón “Crear Caso”.

Lo siguiente que la aplicación mostrará es una nueva ventana en la que tendremos que repetir el mismo procedimiento que en la anterior pero para la creación de un nuevo equipo y la ruta donde se encuentra montado el medio.

Por lo tanto se debe rellenar el campo “Nombre del Equipo”, ya que es obligatorio y añadir al menos un medio de manera análoga a la adición de investigadores a la lista.

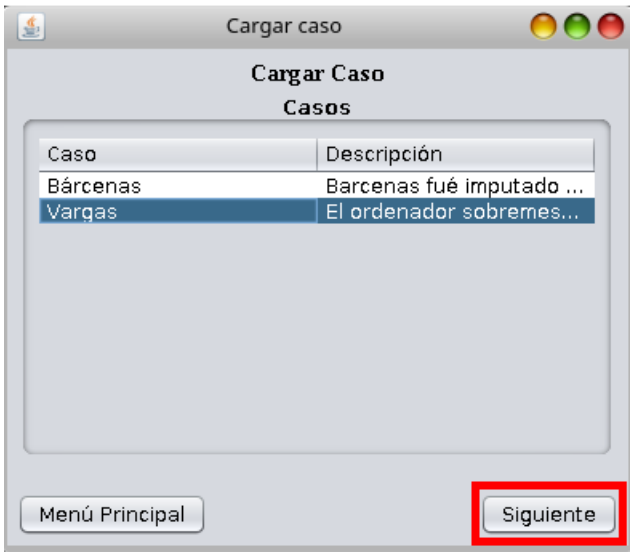


Por ultimo pulse el botón "Crear Equipo", si no aparece ningun mensaje de error el caso se habrá creado correctamente y la aplicación le redirigirá al menú del caso y mostrará las opciones disponibles para el mismo.

2.2. Cargar Caso

Otra de las opciones que podemos elegir en el menú principal es la opción "Cargar caso", que está ubicada en el contenedor de botones relativos a los casos.

Tras esto la aplicación mostrara una lista de los Casos que existen en ese momento, debemos seleccionar uno de ellos pulsando en uno de ellos. Una vez seleccionado el caso pulse el botón "Siguiente".



Tras esto la aplicación le redirigirá al menú del caso y mostrará las opciones disponibles para el mismo.

2.3. Borrar Caso

El último botón que encontramos en el contenedor de botones relativo a los casos, ubicados en el meú principal, es el de "Borrar caso".

Tras pulsarlo la aplicación, nos muestra una lista de los casos en una ventana, de los cuales debemos elegir uno para su eliminación total del sistema.

Se advierte que una vez que se elimina un caso será imposible volver a recuperar los datos borrados, por lo que se ruega proceder con cautela. Para proceder se debe pulsar en el botón "Borrar Caso".

Una vez hecho esto el caso será eliminado y la aplicación le redirigirá al menú principal.

2.4. Crear Grupo Personalizado

Por la parte del contendor de botones referentes a los grupos personalizados, ubicados en el menú principal, la primera opción que nos encontramos es la de "Crear Grupo".

Una vez pulsado este botón el sistema nos mostrará un formulario con varios campos para rellenar.

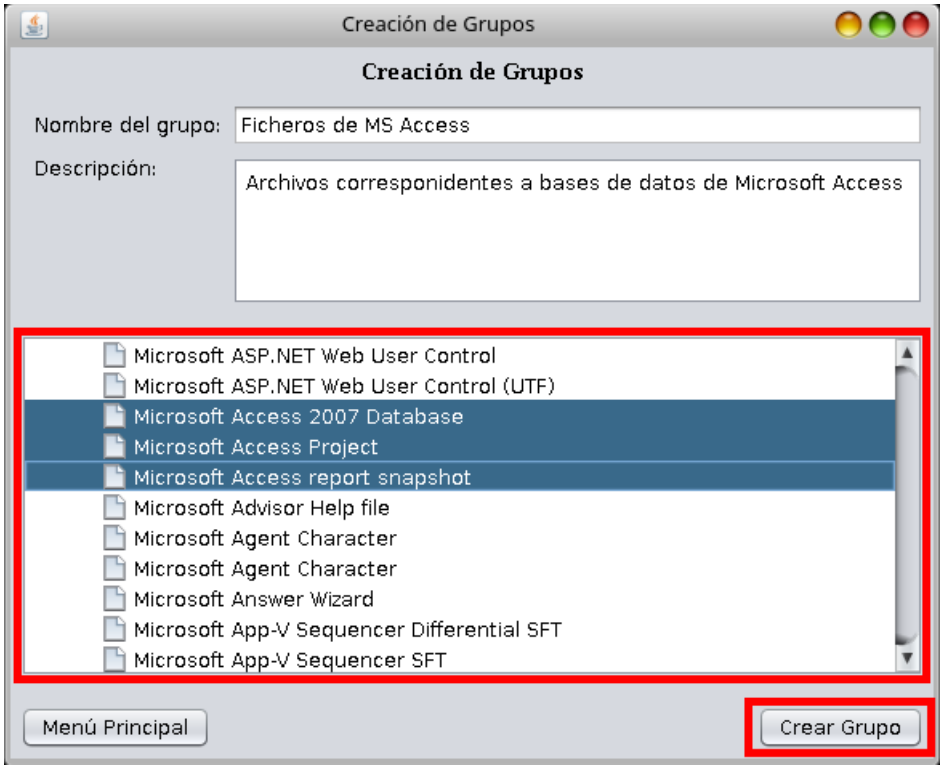
Se debe rellenar al menos el campo "Nombre del Grupo" y debe tener valor, único en caso contrario antes de pasar al siguiente paso el sistema nos mostrará un error y tendremos que cambiarlo.

Opcionalmente podemos rellenar el campo "Descripción", no es obligatorio pero aconsejamos

poner una pequeña descripción del grupo personalizado a crear.

También deberá añadir al menos un fichero de los que se muestran en el arbol de la parte inferior, como son demasiados estan agrupados por orden alfabético, así que para seleccionarlos, primero debe desplegar las letras que vaya a necesitar y luego podrá hacer una selección múltiple.

Cuando haya acabado de seleccionar los ficheros, para finalizar el proceso, pulse en el botón "Crear grupo".



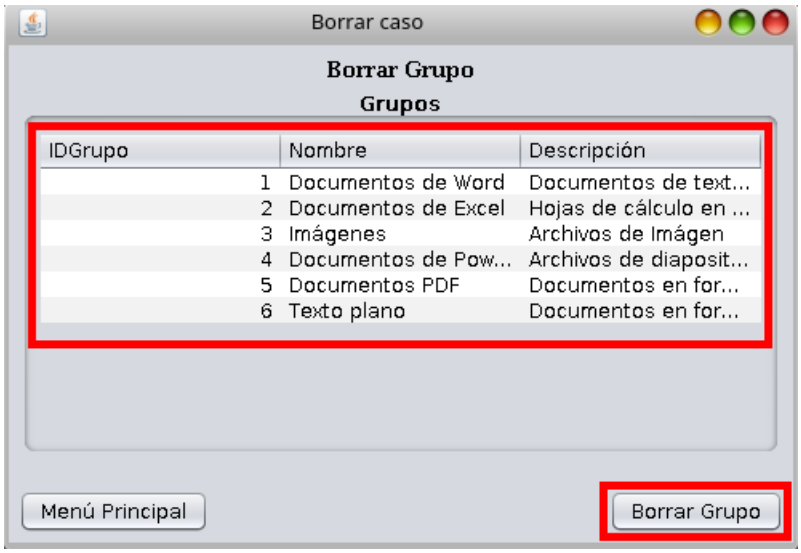
Tras esto la aplicación le redirigirá al menú principal.

2.5. Borrar Grupo Personalizado

El último botón que encontramos en el contenedor de botones relativo a los grupos, ubicados en el meú principal, es el de "Borrar Grupo".

Tras pulsarlo la aplicación, nos muestra una lista de los grupos personalizados en una ventana, de los cuales debemos elegir uno para su eliminación total del sistema.

Se advierte que una vez que se elimina un grupo personalizado, será imposible volver a recuperar los datos borrados, por lo que se ruega proceder con cautela. Para proceder se debe pulsar en el botón "Borrar Grupo".

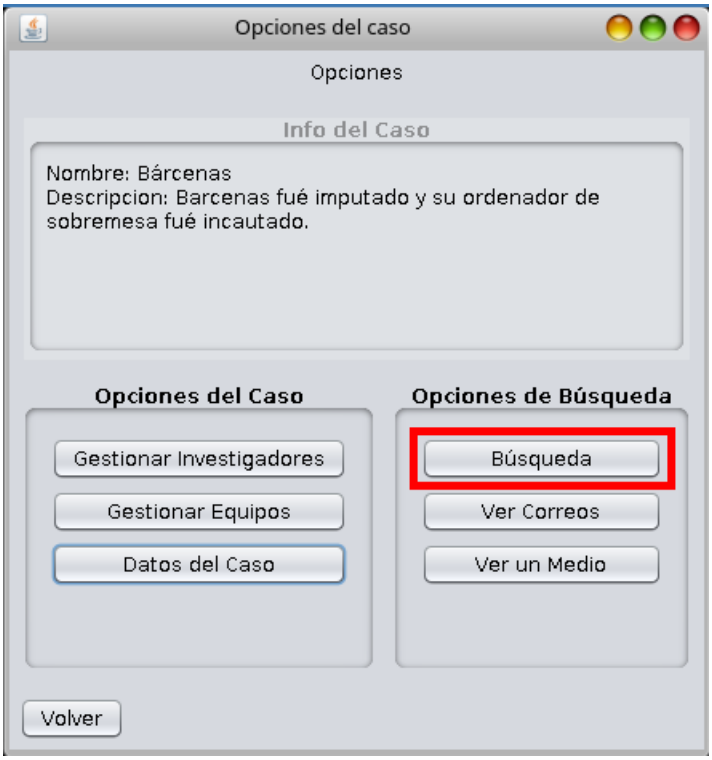


Una vez hecho esto el grupo personalizado será eliminado y la aplicación le redirigirá al menú principal.

2.6. Búsquedas y Copiado de Ficheros

Para hacer búsquedas de ficheros lo primero que tendremos que hacer es cargar un caso siguiendo los pasos de la sección del manual "Cargar Caso".

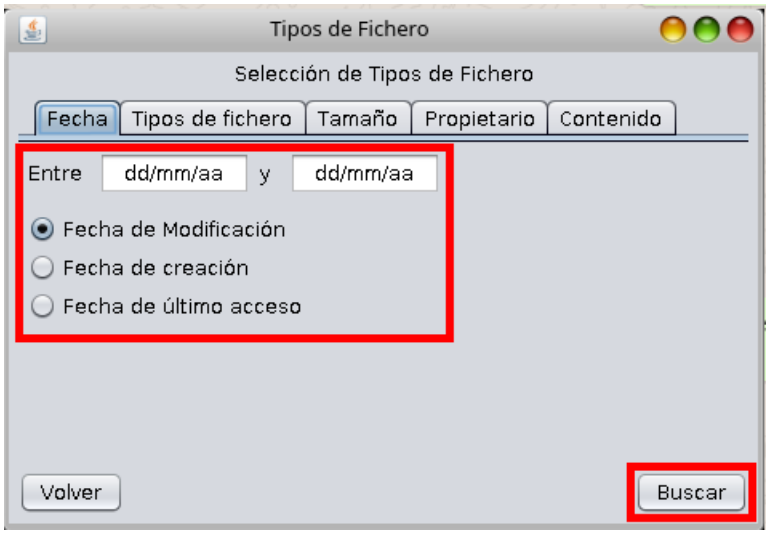
Una vez correctamente cargado el caso debemos pulsar en el boton "Búsqueda", tras esto se mostrará una lista con el equipo y el medio sobre los cuales se va a trabajar, asegurese de que el medio esta conectado y puesto en el punto de montaje correcto, después pulse el botón siguiente.



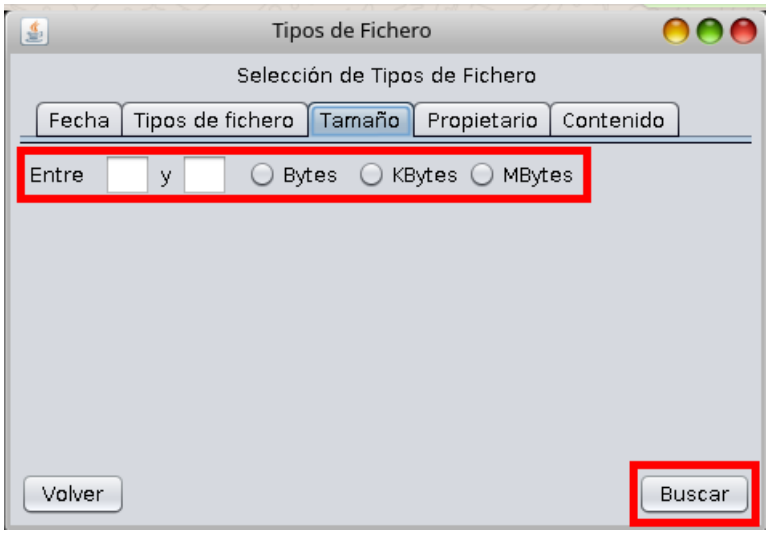
Hecho esto el sistema mostrara la ventana de las búsquedas, en la cual tendremos varias pestañas cada una para un tipo de búsqueda diferente, propietario, fecha, texto, tipo...

Se deberá entrar en una de las pestañas y rellenar **todos** los campos que contenga la pestaña, puesto que todos son obligatorios en este paso, teniendo los siguientes casos:

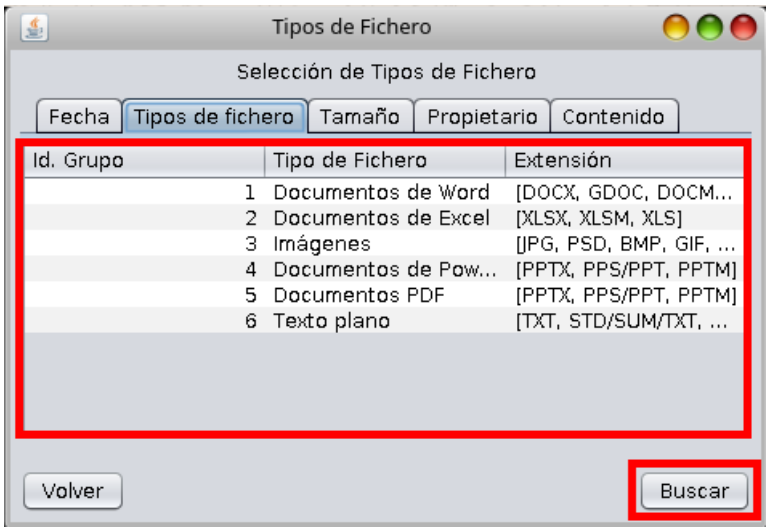
- En el caso de la fecha se selecciona el tipo de fecha (Creación, Acceso o Modificación) y deberá introducir la fecha de inicio de la Búsqueda y la fecha tope.



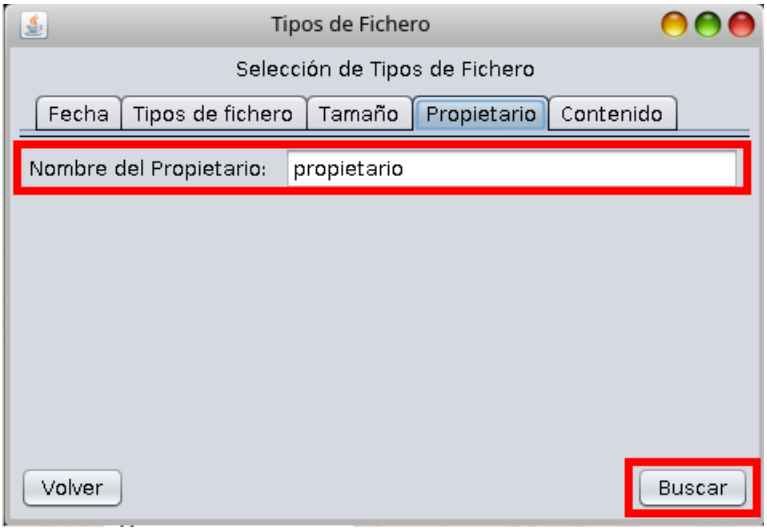
- En el caso del tamaño tendremos que elegir las unidades (B, KB y MB) y tendremos que intoducir un rango de tamaños (ambos inclusive).



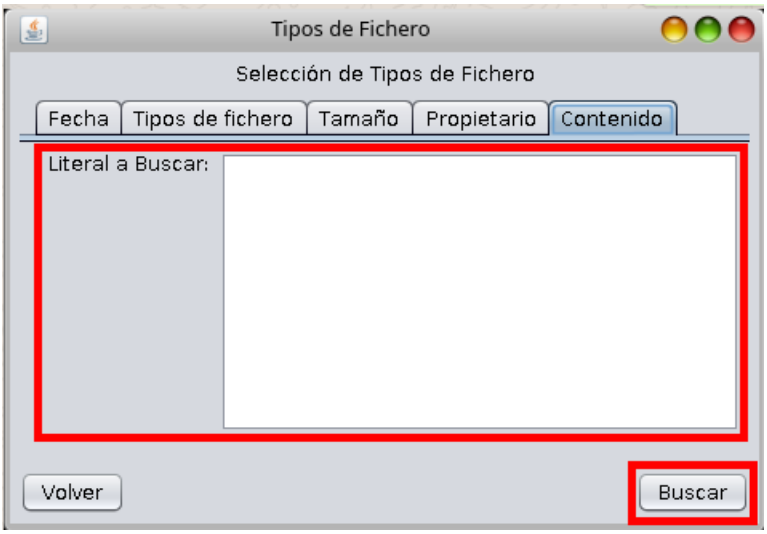
- En el caso de tipos de fichero simplemente tendremos que elegir un grupo personalizado de tipos de ficheros.



- En el caso del propietario debemos introducir el usuario del cual es propietario el fichero.

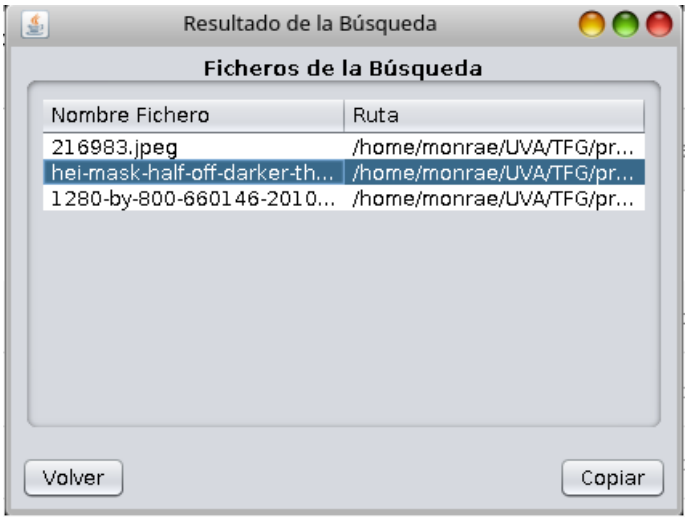


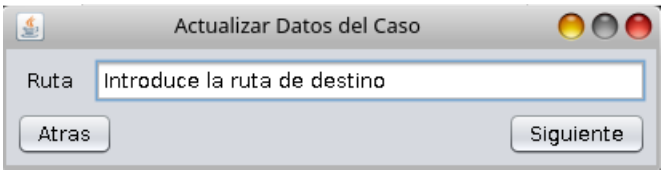
- En el caso del contenido tendremos que introducir el texto a buscar en el interior de los ficheros.



Una vez rellenados los campos o hechas las elecciones pertinentes solo falta pulsar en el botón "Buscar", si hay muchos ficheros en la carpeta esta búsqueda puede tardar unos minutos.

Tras esto el sistema muestra una lista de ficheros que coinciden con los criterios de la búsqueda y se podrá hacer una selección múltiple de aquellos que necesite conservar y copiar en la ruta que se tendrá que introducir justo después de pulsar en el botón "Copiar". Después de introducir la ruta se debera pulsar en el boton "Siguiente".

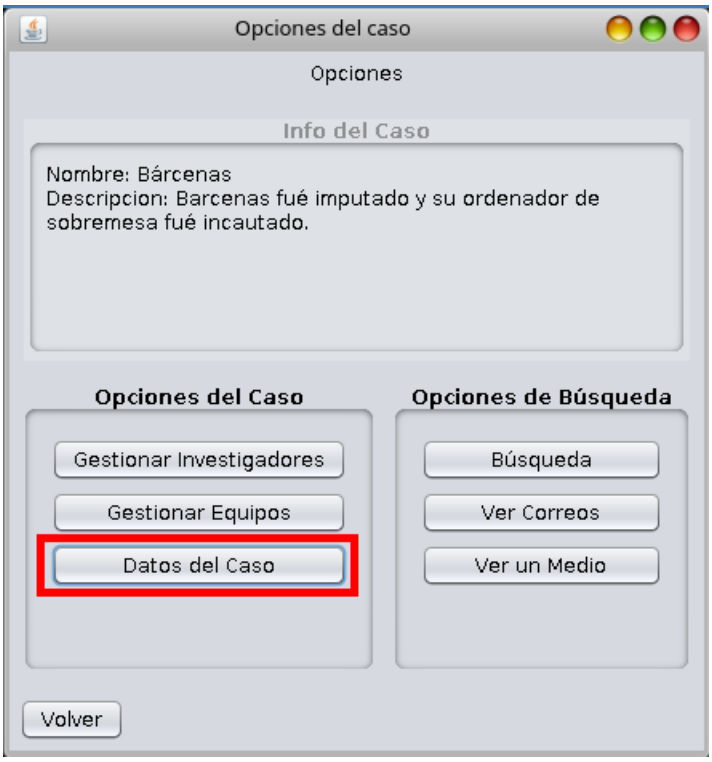




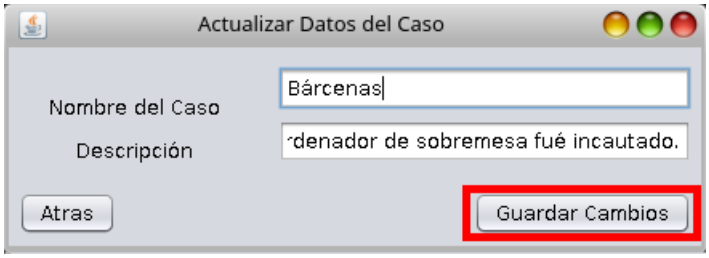
Por último saldrá el resumen de los ficheros copiados y su equivalencia en la carpeta de destino.

2.7. Modificar Datos del Caso

Para modificar los datos de un caso existente lo primero que tendremos que hacer es cargar un caso siguiendo los pasos de la sección del manual “Cargar Caso”.



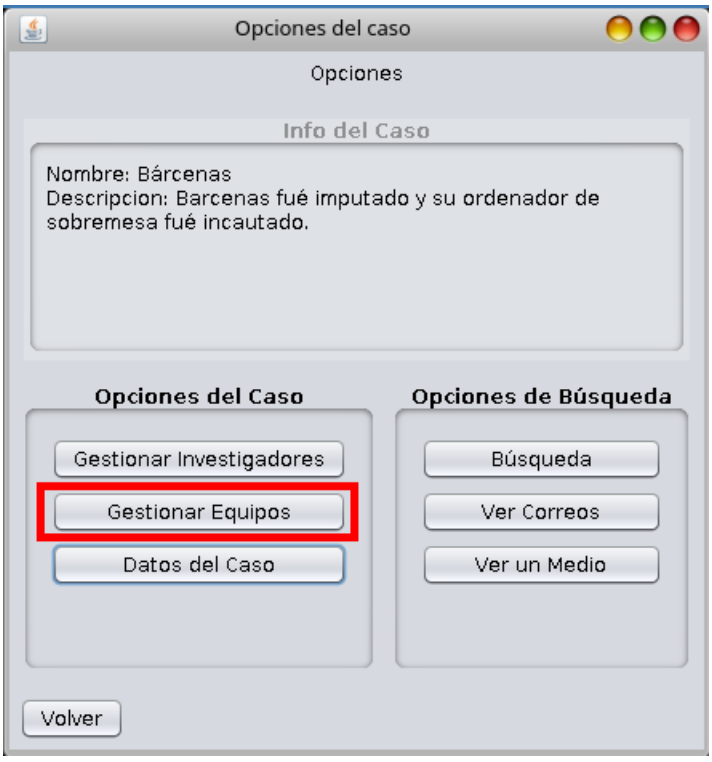
Después tendrá que pulsar el botón “Datos del caso” ubicado en las opciones del caso, tras lo cual saldrá una nueva ventana con dos campos, si desea volver, pulse en el botón “Atras”, si no, cambie cualquiera de los dos campos y pulse en el botón “Guardar cambios”.



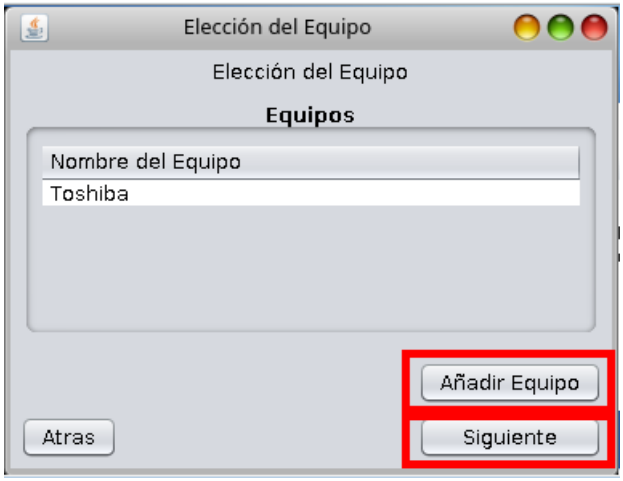
Si alguno de los campos contiene un valor no válido, el sistema mostrará un mensaje de error cuando se pulse en el botón “Guardar cambios”.

2.8. Gestionar los equipos en un Caso

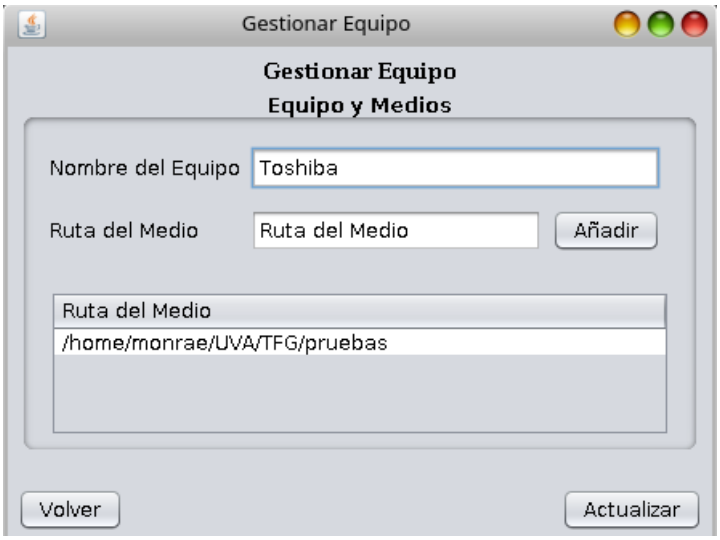
Para gestionar los equipos en un caso lo primero que tendremos que hacer es cargar un caso siguiendo los pasos de la sección del manual “Cargar Caso”.



En las opciones del caso deberá pulsar al botón “Gestionar Equipos”, tras lo cual aparecerá una ventana en la que podrá añadir equipos o modificar los Equipos ya existentes Para Añadir un nuevo equipo pulse en Añadir Equipo, para modificar uno selecciónelo y pulse en el Botón “Siguiente”.

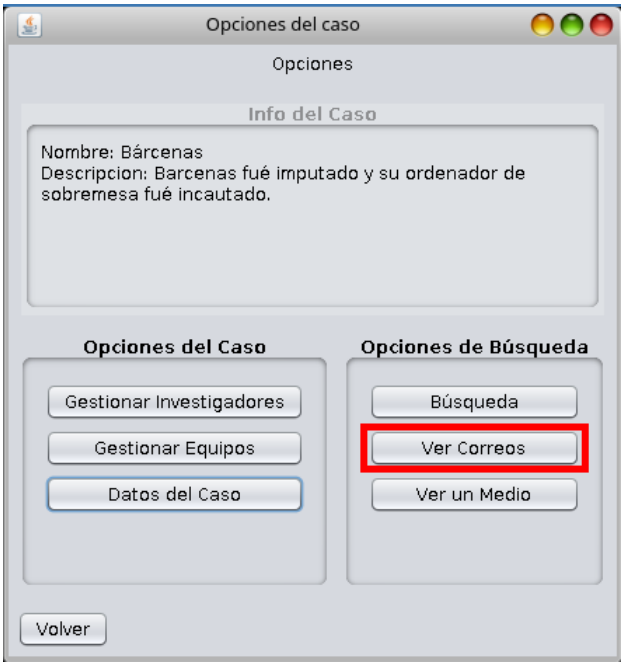


Tanto si Añade, como si modifica un Equipo existente aparecerá la ventana de creación de Equipos en la que tendrá que rellenar (o modificar) el nombre del equipo y al menos añadir un Medio a la lista de Medios poniendo la ruta en el campo "Ruta del medio" y dando en el botón "Actualizar".

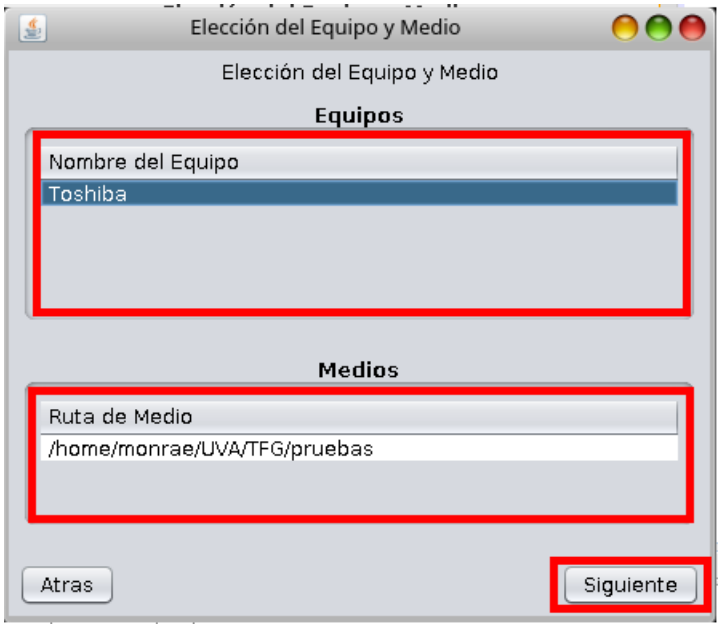


2.9. Ver correos en un Medio

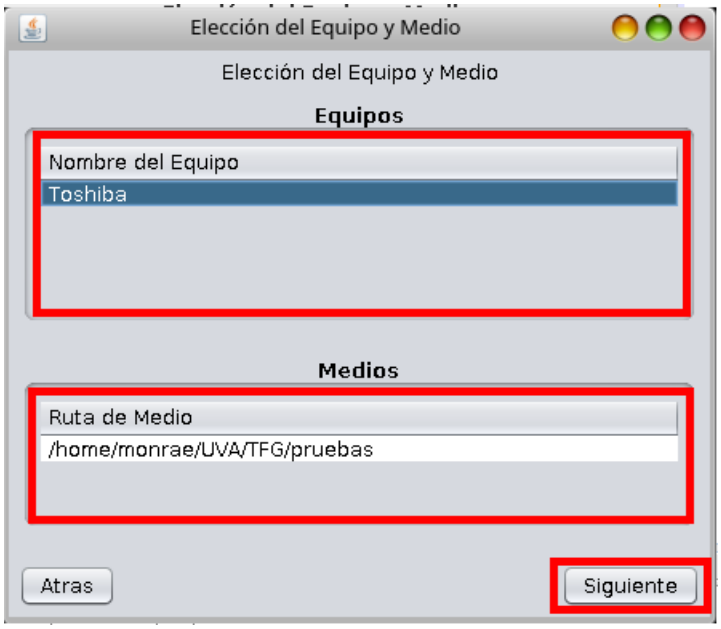
Para ver los correos que existen en un medio lo primero que tendremos que hacer es cargar un caso siguiendo los pasos de la sección del manual "Cargar Caso".



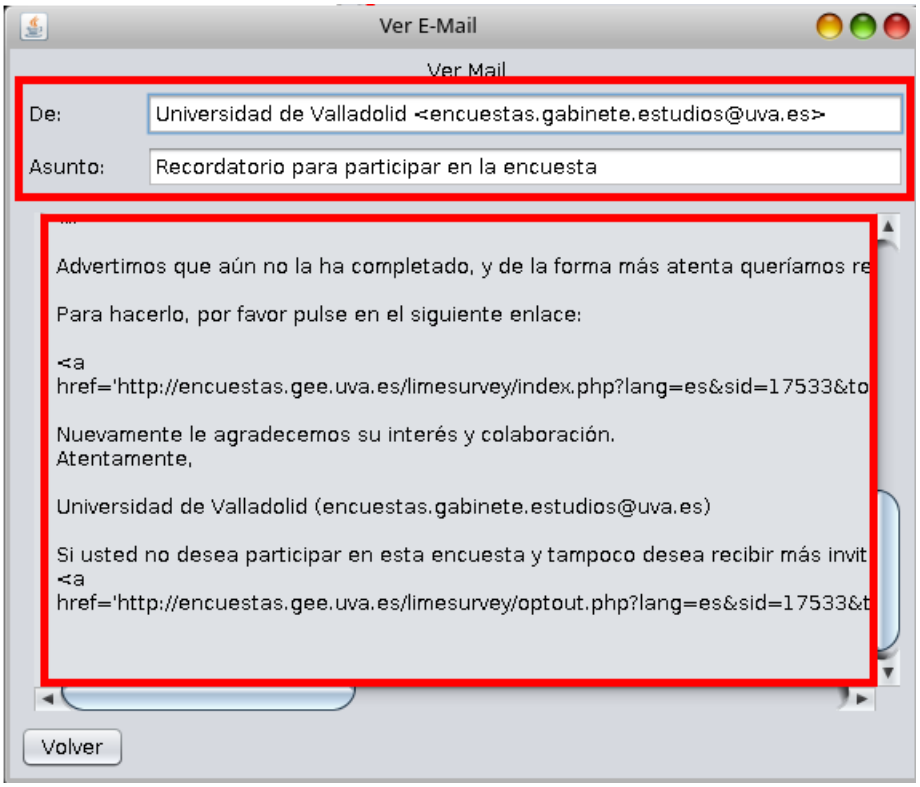
En las opciones del caso pulse en la opción ubicada en el contenedor de botones de Búsquedas, “Ver Correos”, tras lo cual debe elegir el equipo y el medio en el que desea buscar los correos y pulsar en el botón “Siguiente”.



Cuando el sistema muestre una lista de correos existentes en el medio debe seleccionar uno y pulsar en el botón “Ver E-Mail”.



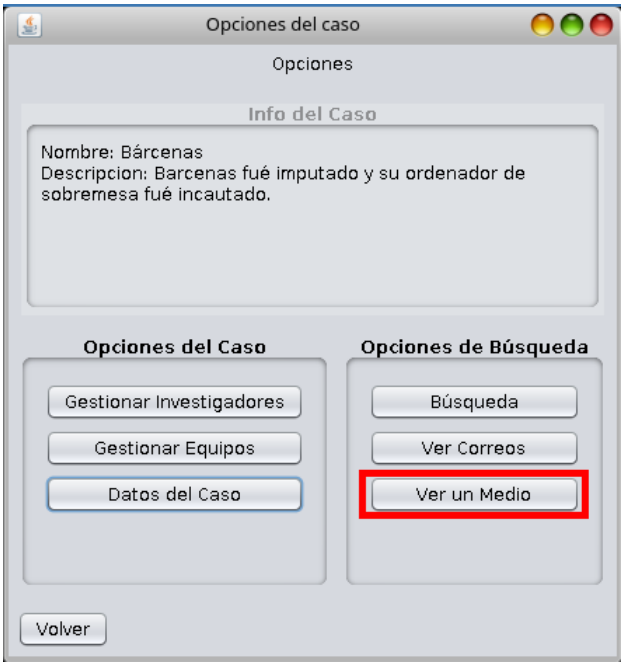
Por último el sistema mostrará un visor con el contenido del E-mail.



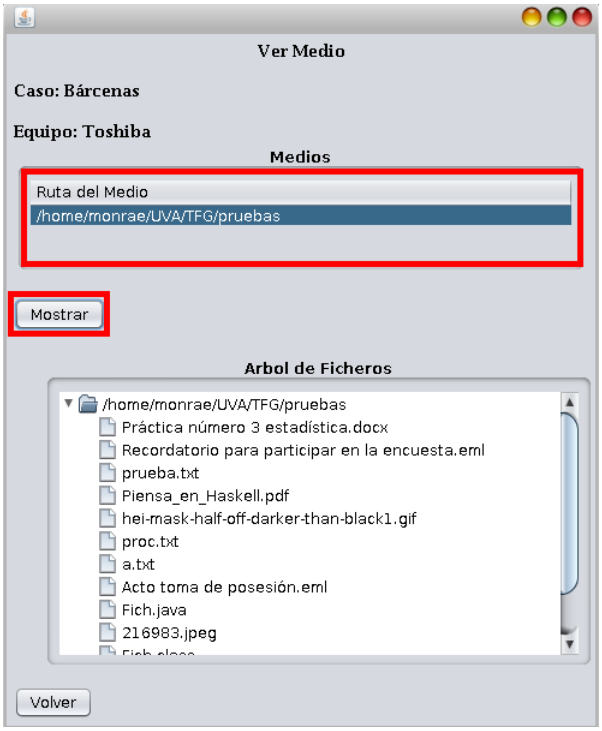
2.10. Mostrar el contenido de un Medio

Para mostrar el contenido de un medio existente en un medio lo primero que tendremos que hacer es cargar un caso siguiendo los pasos de la sección del manual “Cargar Caso”.

En las opciones del caso deberá pulsar en el botón “Ver un Medio” ubicado en el contenedor de botones relativos a las Búsquedas, después debera seleccionar el equipo y pulsar en el botón “Siguiente”, si no selecciona el equipo el sistema le mostrará un mensaje de error y tendrá que repetir este último paso.



Una vez hecho esto seleccionamos el medio deseados en la nueva ventana que nos muestra la aplicación y deberá pulsar el botón "Mostrar".



Por último el sistema le mostrará un arbol con los ficheros y directorios del medio.

2.11. Consultar Informes

Para ello solo se tiene que acceder al fichero SleuthAss.log que se genera automáticamente en la carpeta que contiene el fichero jar de ejecución de la aplicación.

Es un fichero de texto plano que contiene en lenguaje legible por las personas, el conjunto de operaciones que se han ido ejecutando por medio de la aplicación.

Apéndice 3

Glosario

Término	Definición
Cadena de caracteres	En programación, una cadena de caracteres, palabras, ristra de caracteres o frase (string, en inglés) es una secuencia ordenada (de longitud arbitraria, aunque finita) de elementos que pertenecen a un cierto lenguaje formal o alfabeto.
Cadena de custodia	Proceso que verifica la integridad y manejo adecuado de la evidencia.
Caso	Palabra genérica que hace referencia a una situación, suceso, acontecimiento, conjunto de circunstancias, etc., entendidos como una entidad particular y diferenciada, en nuestro caso se entiende como un expediente abierto a una investigación sobre un delito.
Etiqueta	Una etiqueta o tag es una palabra clave asignada a un dato almacenado en un repositorio. Las etiquetas son en consecuencia un tipo de metadato, pues proporcionan información que describe el dato.
Interfaz	En informática, se utiliza para nombrar a la conexión funcional entre dos sistemas, dispositivos o componentes de cualquier tipo, que proporciona una comunicación de distintos niveles permitiendo el intercambio de información entre ellos.
Fichero	Conjunto de bits que son almacenados en un dispositivo. Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene.
Metadatos	Un grupo de metadatos se refiere a un grupo de datos que describen el contenido informativo de un objeto al que se denomina recurso.

Término	**Definición**
Soporte de Almacenamiento de datos	El soporte de almacenamiento de datos es el material físico donde se almacenan los datos que pueden ser procesados por una computadora, un dispositivo electrónico, o un sistema informático.
Ocurrencia	Aparición de un elemento lingüístico en un texto.
Prefijo	El prefijo es un morfema de la clase de los afijos que se antepone a una raíz o lexema para hacer una forma léxica con diferente significado.
Sufijo	El sufijo es un morfema de la clase de los afijos que se añade al final de una palabra o de su raíz para formar una palabra derivada.

Tabla 3.1: Glosario de términos.

Bibliografía

[1] B. Nelson,"Guide to Computer Forensics and Investigations"

[2] Craig Larman, "UML y Patrones", 2ª Edición

[3] E. Casey, "Handbook of Digital Forensics and Investigation"

[4] E. Gamma & R. Helm & R. Johnson & J. Vissides, "Design Patterns"

[5] G. Booch "Análisis y diseño orientado a objetos con aplicaciones"

[6] Gonzalo Navarro & Mathieu Raffinov, "Flexible Pattern Matching In Strings"

[7] P. Sznajdleder, "Java a Fondo Estudio del lenguaje y desarrollo de aplicaciones"

[8] "A Guide of Project Management Body of Knowledge", 5ª Edción

Enlaces Web

[1] Algoritmos de busqueda de subcadenas, https://es.wikipedia.org/wiki/Algoritmos_de_busqueda_de_subcadenas

[2] Java Development Kit, https://es.wikipedia.org/wiki/Java_Development_Kit

[3] Metadatos, https://es.wikipedia.org/wiki/Metadato

[4] Magic Number, https://es.wikipedia.org/wiki/Numero_magico

[5] NetBeans, https://es.wikipedia.org/wiki/NetBeans

[6] Tutorial de Apache Derby, https://db.apache.org/derby/papers/DerbyTut/index.html