Desarrollo de herramienta para análisis de cabeceras de correos electrónicos

Enzo Notario, Esteban Rivetti, Beatriz Parra de Gallo
IEstIIng – Facultad de Ingeniería
Universidad Católica de Salta
Salta, Argentina
enzo.notario@gmail.com, erivetti83@gmail.com, bgallo@ucasal.edu.ar

Abstract

El presente trabajo describe una solución automatizada al problema de extraer la información que se encuentra en una cabecera de un correo electrónico para su posterior análisis forense. Dicha cabecera se presenta en forma de texto plano, lo cual dificulta la localización de ciertos datos que puedan ser relevantes para un análisis forense, particularmente cuando se trata de un conjunto de correos provenientes de una única cuenta. Se detalla la secuencia de pasos elaborada para el proceso ETL (Extracción, Transformación y Carga) ajustado a criterios de repetibilidad de obtención de la prueba, que se exigen a las técnicas científicas para el análisis forense.

.

1. Introducción

Un correo electrónico se compone por una cabecera y un cuerpo presentados en forma de texto plano, separados por una línea en blanco y suelen ajustarse al formato definido por el RFC 822[1], aunque algunos servidores de correos modifican dicho formato para atender a sus necesidades. Los datos que contiene la cabecera de un correo pueden ser muy importantes en el análisis forense, que deben ser separados y tratados para conformar una base de datos útil.

Tomando como base el trabajo "Estudio comparativo de desempeño de herramientas para el análisis forense de correos electrónicos" [1], se desarrolla una solución para lograr obtener la información necesaria y responder los interrogantes que usualmente se hacen en una pericia.

El presente trabajo consiste en tomar una muestra de 5281 correos electrónicos que se extrajeron de una cuenta de correo y se exportaron como archivos independientes, separar la cabecera del cuerpo y recorrer cada línea de la cabecera para ir recolectando la información deseada. Posteriormente se utilizan para poblar una Ontología, la cual forma la base de conocimientos para apoyar el análisis forense de los correos electrónicos.

[1]Se puede consultar en: https://www.ietf.org/rfc/rfc822.txt

El sistema se realizó utilizando el framework Laravel [2] para el tratamiento del correo electrónico, Apache Jena [3] para la construcción de la Ontología y Apache Fuseki [4] como servidor SPARQL.

Este trabajo se organiza de la siguiente manera: el apartado 2 aborda una breve descripción de la ontología para el análisis forense de correos electrónicos. La sección 3 sintetiza el procesamiento del correo electrónico, en la sección 4 se explica cómo llevar a cabo la herramienta la separación del encabezado y el cuerpo del correo, en el apartado 5 se hace referencia a la extracción de los campos clave, la sección 6 explica el envío de instancia a Apache Jena y por último la sección 7 describe las conclusiones arribadas.

2. Ontología para el análisis forense de un correo electrónico

El análisis forense no debe presentarse como un reporte técnico sino como información sistemática y con sentido semántico en el marco de la causa judicial.

En el trabajo citado los autores plantean como objetivo el de contar con un marco de referencia común apoyándose en tecnologías semánticas, enfocándose en la trazabilidad de un correo electrónico, es decir, en todo el camino que recorre un correo electrónico hasta llegar a su destino final. Para ello es necesario contar con los datos relevantes para este análisis, como ser fecha, direcciones ip, nombre de cuentas entre otros.

En particular, la solución propuesta permite extraer de forma simple y ágil la información que se necesita para conformar la base de conocimientos.

La información clave a extraer de los correos electrónicos, se derivan del trabajo "Hacía una Ontología para el soporte de la trazabilidad del correo electrónico en la Forensia Digital" [5], los cuales buscan responder los interrogantes de un punto de pericia que usualmente se solicitan en un análisis forense de un correo electrónico.

Los interrogantes a los que hacemos mención en el párrafo anterior, se pueden buscar en los puntos de pericia que usualmente se proponen al solicitar un análisis forense de un correo electrónico.

- ¿Cuál es la fecha, hora y dirección IP de emisión del correo electrónico?
- ¿Cuál es la fecha, hora y dirección IP de recepción del correo electrónico?
- ¿Cuál es el nombre de usuario y dirección de e-mail del Autor del mismo?
- ¿Cuál es el nombre de usuario y dirección de e-mail del Receptor del mismo?
- ¿Es posible establecer la trazabilidad del mensaje desde que se envía hasta que se recibe?
- ¿Cuáles son los diferentes actores/servicios que participaron de la transmisión?

Por razones de espacio no se incluye la ontología completa, pero a modo de síntesis, la Fig. 1 señala las clases intervinientes en el proceso de instanciación de la ontología a partir de la carga automática señalada.

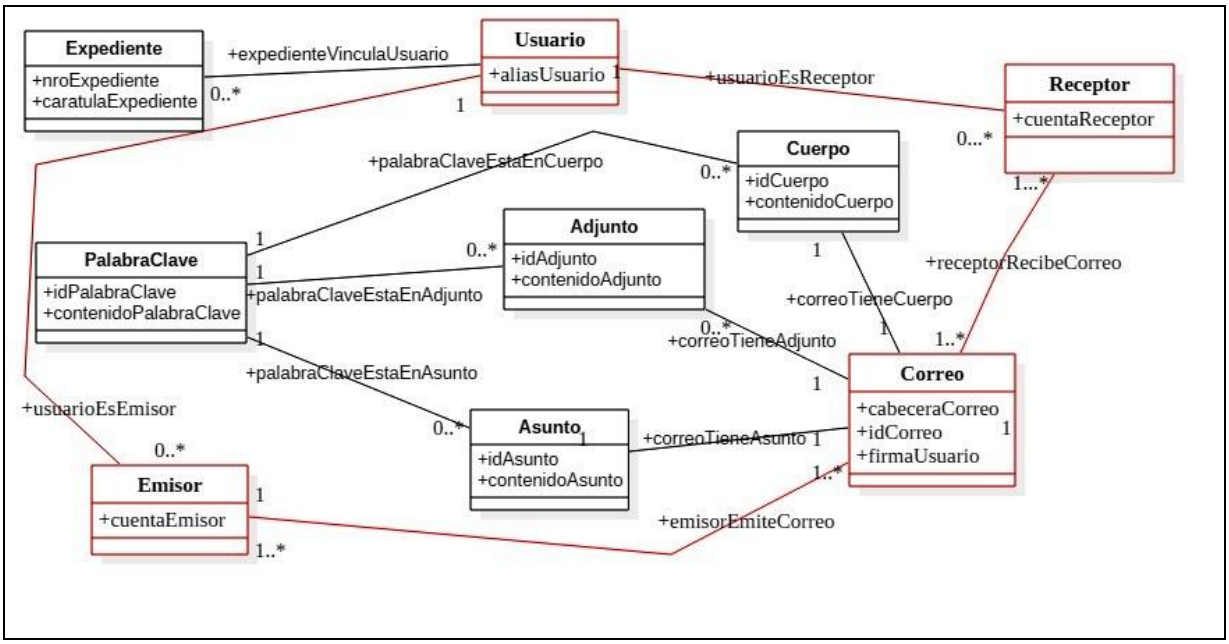

Figura 1 Vista parcial de la Ontología

3. Procesamiento de un correo electrónico

Un correo electrónico pasa por diferentes procesos hasta conformar una instancia en la ontología: se separa la cabecera del cuerpo, se extraen los atributos relevantes y luego se envía la información obtenida a Apache Fuseki para que sea incluida en la ontología.

Estos procesos se muestran en la Fig.2, mientras que la **Figura 3**3 describe la arquitectura de procesamiento utilizada.

Para realizar el procesamiento de los 5281 correos electrónicos, se realizaron los pasos que a continuación se detallan, prestando especial atención a los principios de resguardo de la privacidad y confidencialidad de los datos.

Paso 1: Extracción de los correos en formato EML

Para la experiencia se utilizó una cuenta de correo de dominio *gmail*, de la cual se tomarían los correos para la instanciación de la ontología. En la computadora de configuración standard, que se utilizó para la experiencia, se procedió a:

1) Instalación del gestor de correo Thunderbird Versión 52.6.0 (32-bit) y en él se habilitó la cuenta de la que se tomaron los datos de prueba.

2) Cuidando de inhabilitar la conexión a internet para no acceder a la cuenta real, se realizó el proceso de exportación de los correos a *formato EML*, mediante las herramientas de importación/exportación de cuentas de la aplicación

3) La aplicación generó un directorio de archivos con esta estructura:

- una página HTML con un cuadro descriptivo de los correos
- un subdirectorio con tantos archivos como correos se exportaron

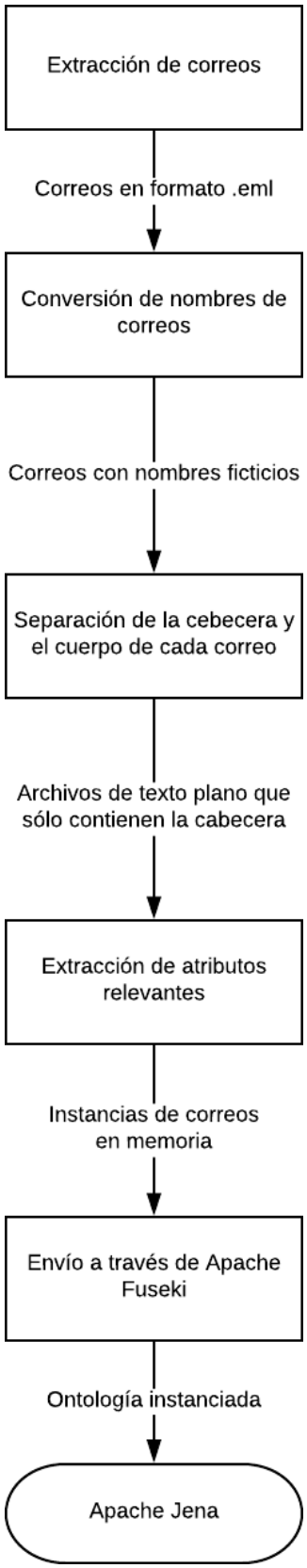

Figura 2: Procesos del sistema

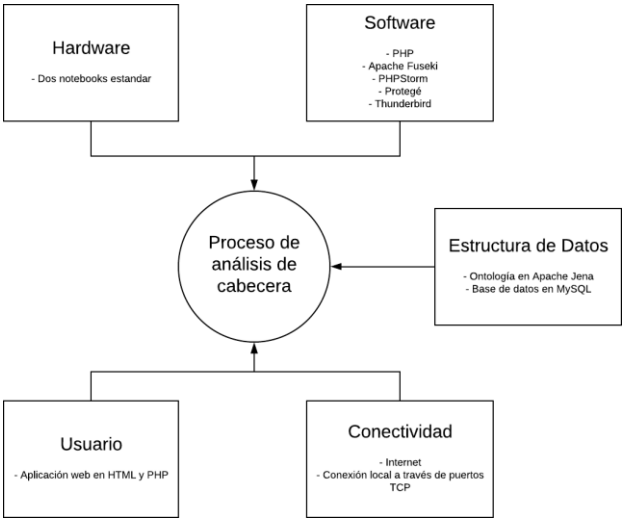

Figura 3: Arquitectura del sistema

Paso 2: Conversión de los nombres de correo

A fin de preservar los datos de identificación de las cuentas de correo contenidas en el banco de pruebas obtenido, se decidió reemplazar cada cuenta real por un nombre ficticio, según el siguiente procedimiento:

2.1) Por cada correo, se analizó su cabecera como se explica más adelante en el capítulo 5 (página 5) pero sólo para extraer las direcciones de correo que contienen, almacenándolas en una base de datos MySQL, específicamente en una tabla *"accounts"*, que tiene el esquema que se muestra en Fig. 4. Luego se copiaron y pegaron en una hoja de cálculo.

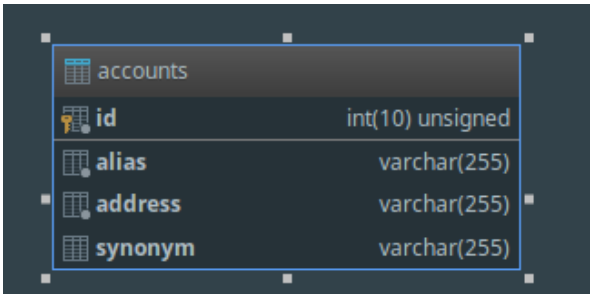

Figura 4: Tabla "accounts"

2.2) Se generaron 4 columnas con el siguiente contenido:

- COLUMNA A: contiene la cuenta de correo obtenida del banco de pruebas
- COLUMNA B: la expresión "usuario1" definido luego como serie numérica para que al barrera hacia abajo, se generara el valor en la fila siguiente a partir de un incremento natural (usuario2, usuario3, ...)
- COLUMNA C: la expresión "@" valor constante en toda la columna
- COLUMNA D: la expresión "dominio1" definido en idénticos términos que la primera columna, para generar

datos con un incremento natural (dominio1, dominio2, ...)

- COLUMNA E: la exrepsión ".com" valor constante en toda la columna
- COLUMNA F: función de concatenación de B+C+D+E

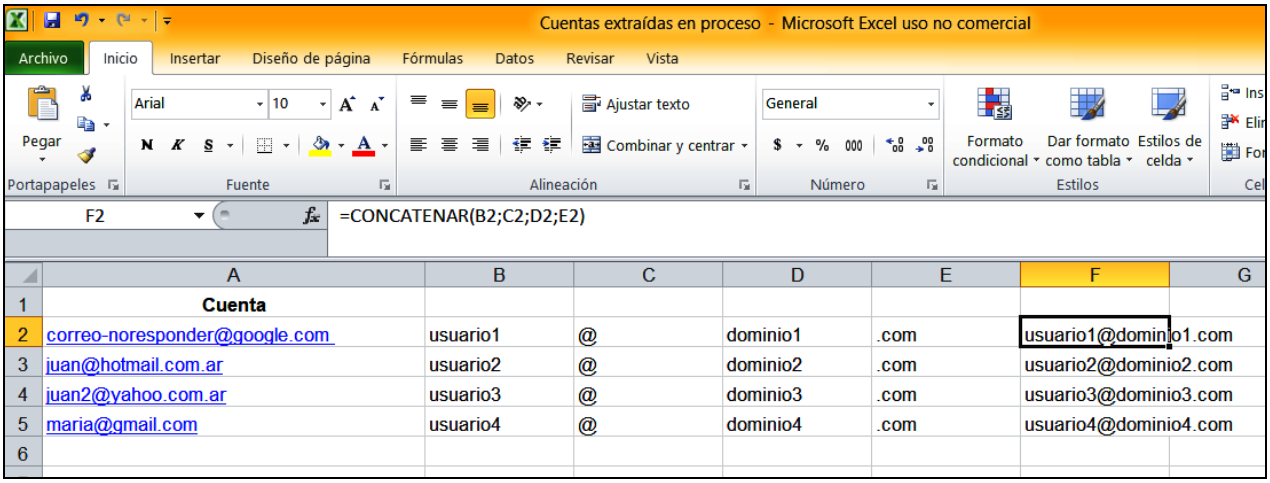

Figura 5: Conversión de nombres de cuentas de correo

2.3) se copiaron las celdas de la fila 1 hasta la última fila para generar el sinónimo en los 1610 correos diferentes que tenía la planilla.
2.4) se convirtió la columna F en un valor fijo
2.5) se eliminaron las columnas B,C y D para obtener una tabla con 2 valores: la cuenta original y la cuenta ficticia correspondiente a esa cuenta original.

```
public function run()
{
   $arr = json_decode(\Storage::disk('synonyms')->get('json.json'),
true);

   foreach ($arr as $key => $value) {
      $address = trim($value['address']);
      $synonym = trim($value['synonym']);

      $account = \App\Http\Models\Account::where('address', '=',
$address)->first();

      if (! $account) {
         $account = \App\Http\Models\Account::create([
            'alias' => $address,
            'address' => $address,
         ]);
      }

      $account->synonym = $synonym;

      if ($account->alias == $account->address) {
         $account->alias = $synonym;
      }

      $account->save();
   }
}
```

Figura 6: Código utilizado para importar los sinónimos

2.6) La hoja de cálculo resultante fue abierta en *Google Spreadsheets*[6] para exportarla a un formato JSON utilizando el complemento *ExportSheetData*[7]. Este archivo es leído desde Laravel, donde es recorrido y, por cada cuenta, se almacena el sinónimo que se define en la hoja de cálculo. El código utilizado se muestra en Figura 66.

Luego, durante la ejecución del sistema, cada dirección de correo que se encuentre será reemplazada por su sinónimo, que se obtiene utilizando el código que se muestra en Figura 77.

```
public function replace($alias, $address)
{
   $account = Account::where('address', '=', $address)->first();

   if ($account) {
      $alias = $account->alias;
      $address = $account->synonym;
   }

   return [$alias, $address];
}
```

Figura 7: Código utilizado para obtener el sinónimo de una dirección de correo electrónico

Paso 3: Separación de la cabecera y el cuerpo de un correo electrónico

En principio lo único que se obtiene de un correo electrónico es un archivo de texto plano en donde se encuentra tanto la cabecera como el cuerpo. La información que el sistema busca obtener se encuentra en la cabecera, por lo tanto es necesario poder separarlos de manera automatizada.

El sistema es capaz de recibir una carpeta en donde se encuentran los archivos que contienen los distintos correos electrónicos y separar la cabecera del cuerpo de cada correo, guardando la cabecera en otra carpeta para

su posterior tratamiento. Esto se logra siguiendo el formato definido por el RFC 822, que indica que la cabecera y el cuerpo deben estar separados por una línea en blanco. Teniendo en cuenta esto, resulta sencillo plasmarlo en código, como se muestra en **Figura 8**8.

La función *fromFileContent* recibe el contenido del correo electrónico. Con una expresión regular se obtienen las distintas líneas a través de la función *preg_split*. Se recorren las líneas y se las agrega a la variable *$header* hasta que se encuentre una línea en blanco. En tal caso el recorrido se detiene y se retorna la variable *$header* en forma de *string* separado por un salto de línea.

```
public static function fromFileContent(string $content): string
{
    $header = collect();

    $lines = preg_split("/((\r?\n)|(\r\n?))/", $content);

    foreach ($lines as $line) {
        if (trim($line) == '') {
            break;
        }

        $header->push($line);
    }

    return $header->implode("\n");
}
```

Figura 8: Código para separar la cabecera del cuerpo del correo

Una particularidad del sistema es su protección a la privacidad. Cada dirección de e-mail es reemplazada por una aleatoria, que se obtiene utilizando la librería Faker[8], al igual que el asunto, que es reemplazado por un texto aleatorio de hasta doscientos caracteres.

Paso 4: Extracción de atributos relevantes

Una vez separada la cabecera del cuerpo del correo electrónico, se procede a analizarla para extraer la información deseada, que se muestra en la **Figura 9**9.

Atributo	Equivalente a
Subject	Asunto
Message-id	Identificador
Date	Fecha
From	Emisor/es
To	Receptor/es
Received/X-Received	Ocurrencia de transmisión[2]

Figura 9: Atributos relevantes

Para esto se recorre línea por línea la cabecera buscando los distintos atributos. El RFC 822 define un formato para los distintos atributos: cada línea de la cabecera debe empezar con el atributo que define, seguido de dos puntos y luego el valor en sí. Además, se detectó que el valor de algunas líneas en realidad están conformada por varias líneas, las cuales comienzan por al menos un espacio en blanco. Entonces, las líneas que comienzan por un espacio en blanco son consideradas continuación del valor de la anterior.

```
/**
 * @param $line
 *
 * Obtiene el atributo y valor definido en `$line`.
 *
 * @return string
 */
public function parseLine($line)
{
    $key = trim(substr($line, 0, strpos($line, ':')));
    $value = trim(substr($line, strpos($line, ':') + 1));

    // A partir de la línea actual, se obtiene el valor de las siguientes
    // mientras estas comiencen con un espacio en blanco. Esto quiere
    // decir que en realidad son una continuación de la línea actual.
    $subIdx = $lineIdx;
    while ($this->startsWithSpace($this->getNextLineFromIdx($subIdx))) {
        $value .= ' ' . trim($this->getNextLineFromIdx($subIdx));
        $subIdx++;
    }

    return [$key, $value];
}

/**
 * @param $heystack
 *
 * Determina si `$heystack` comienza con al menos un espacio en blanco
 * o un tabulador.
 *
 * @return bool
 */
private function startsWithSpace($heystack): bool
{
    return starts_with($heystack, ' ') || starts_with($heystack, "\t");
}

/**
 * @param $idx
 *
 * Devuelve el valor de la siguiente línea a partir de `$idx`.
 *
 * @return string
 */
private function getNextLineFromIdx($idx): string
{
    $idx++;

    if (count($this->lines) == $idx) {
        return '';
    }

    return $this->lines[$idx];
}
```

Figura 10: Código que obtiene el atributo y valor de una línea de la cabecera

[2] Las *Ocurrencias de Transmisión* son las sucesivas copias del correo electrónica que van depositándose en los servidores de correo que se utilizan durante el proceso de transmisión, y que marcan la trazabilidad del correo.

La función que obtiene el atributo y valor de una línea se muestra en la **Figura 10**. El valor que se obtiene es tratado según el atributo que define.

Se debe tener en cuenta que la manera correcta de leer una cabecera es de abajo hacia arriba, ya que a medida que el correo electrónico va pasando por distintos servidores de correo, estos agregan información al inicio de la cabecera.

Una vez recorridas todas las líneas, se puede conformar una instancia como se muestra en la **Figura 11**.

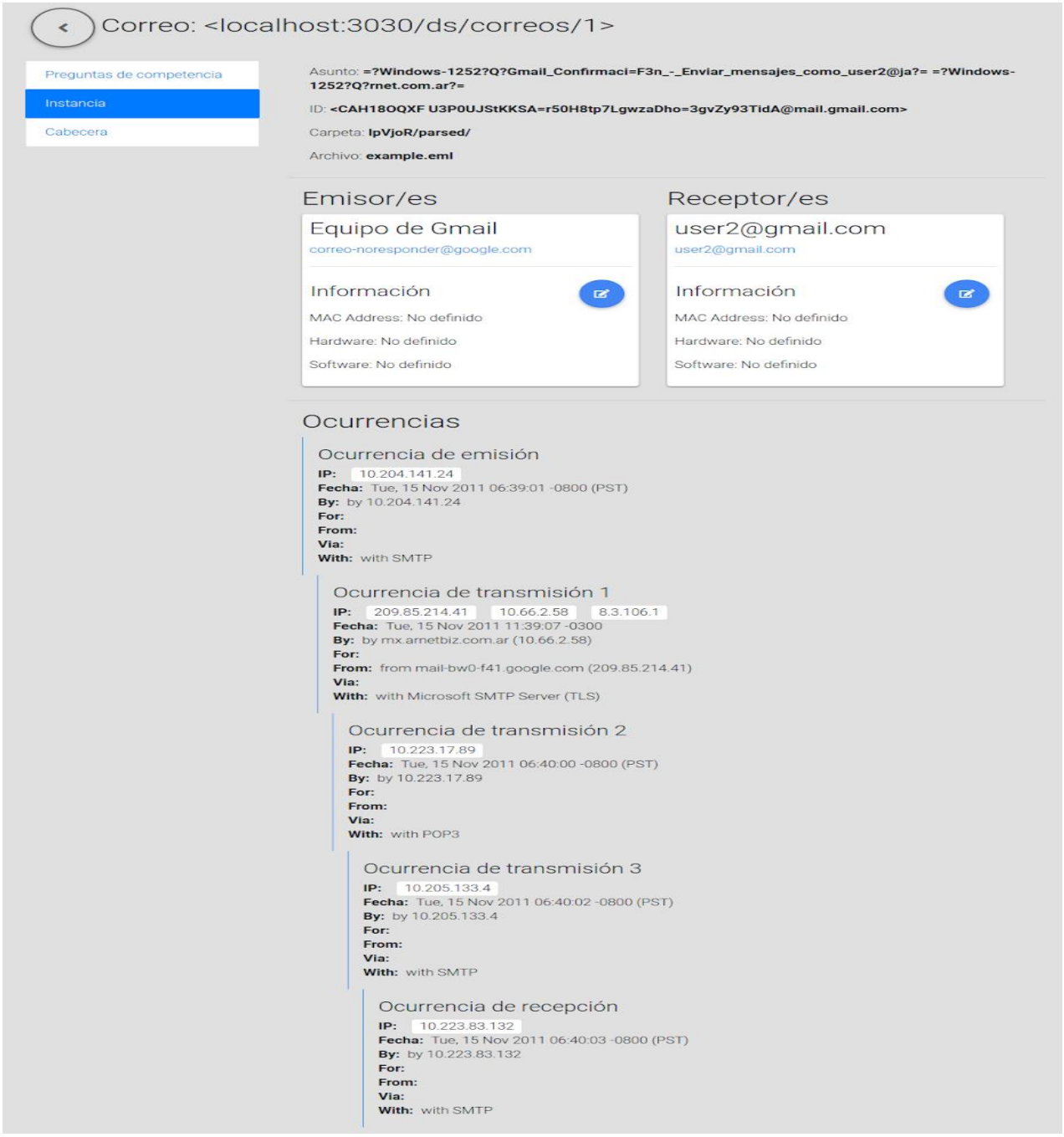

Figura 11: Instancia de un correo

Paso 5: Envío de instancia a Apache Jena

Con la información que se obtuvo se conforma una instancia que será enviada a Apache Jena para construir la ontología. Esto se hace a través de la API que

proporciona Apache Fuseki, con el método POST al endpoint <host>:<port>/<dataset>. Una query de inserción es similar a la que se muestra en la **Figura 12**, y la petición HTTP que se le debe enviar a Apache Fuseki es similar a la que se muestra en la **Figura 13**.

```
PREFIX :
<http://www.semanticweb.org/beatr/ontologies/2018/0/ontologia_correos#>

INSERT DATA {
  GRAPH <localhost:3030/ds/correos> {
    <localhost:3030/ds/correos/1> :asunto 'Asunto del correo' .
    <localhost:3030/ds/correos/1> :id_correo
'<CAH18OQXF+U3P0UJStKKSA=r50H8tp7LgwzaDho=3gvZy93TidA@mail.g
mail.com>' .
  }
}
```

Figura 12: Query SPARQL de inserción

Por último, la Fig.13 muestra la petición *http* de inserción enviada a Apache Fuseki en el último paso del proceso.

Método	Endpoint	Data
POST	localhost: 3030/ds	update=PREFIX+%3A+%3Chttp%3A%2F%2Fwww.semanticweb.org%2Fbeatr%2Fontologies%2F2018%2F0%2Fontologia_correos%23%3E+%0A%0AINSERT+DATA+%7B%0A++GRAPH+%3Clocalhost%3A3030%2Fds%2Fcorreos%3E+%7B+%0A+++++%3Clocalhost%3A3030%2Fds%2Fcorreos%2F2%3E+%3Aasunto+'Asunto+del+correo'+.%0A++++%3Clocalhost%3A3030%2Fds%2Fcorreos%2F2%3E+%3Aid_correo+'%3CCAH18OQXF%2BU3P0UJStKKSA%3Dr50H8tp7LgwzaDho%3D3gvZy93TidA%40mail.gmail.com%3E'+.%0A++%7D%0A%7D

Figura 13: Petición HTTP de inserción enviada a Apache Fuseki

A medida que la ontología se va enriqueciendo de distintas instancias, es posible formular preguntas de competencia como la que se muestra en la Figura 14.

Resultados

El proceso descripto permite la generación del banco de pruebas de la ontología que se está construyendo para el análisis forense de correos electrónicos.

El proceso de extracción, transformación y carga de las cabeceras de correos electrónicos cumple con los requisitos metodológicos necesarios para replicar la experiencia, condición ésta que le asigna rigor científico al proceso pericial.

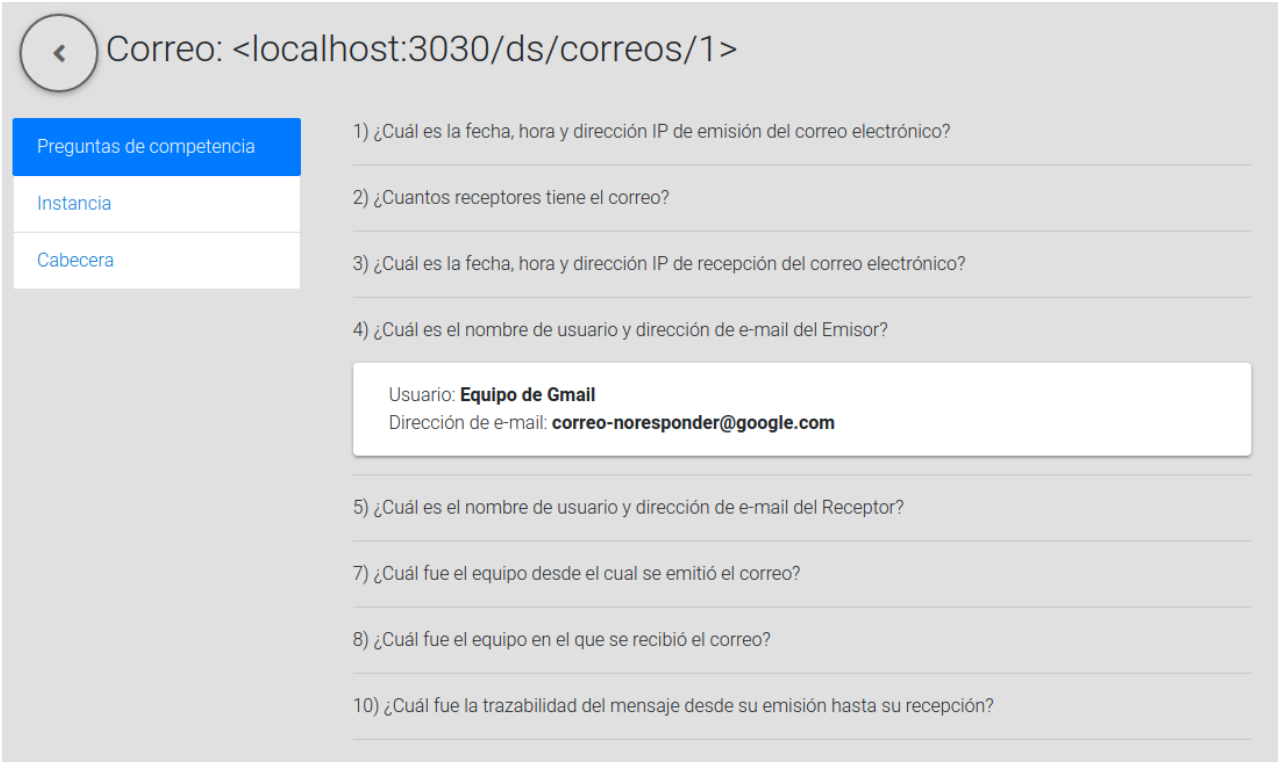

Figura 14: Preguntas de competencia para un correo

Conclusiones

Cabe destacar que el sistema completo se ha construido utilizando tecnologías web, lo cual permitiría disponer de una herramienta en línea capaz de ser utilizada desde cualquier dispositivo con conexión a internet a través de un navegador web.

Las herramientas del mercado, tienen ciertas limitaciones y no responden a todas las preguntas o si lo hacen no es de forma integral. La mayoría de las herramientas son con licencia y tienen limitaciones en la cantidad de correos procesados, lo cual es un impedimento si se desea analizar una cantidad importante de correos.

La herramienta desarrollada, se puede ir adaptando y agregando información de las cabeceras lo cual significa que es totalmente adaptable a las necesidades. La perfomance de la herramienta en analizar cantidades considerables de correos es muy efectiva en comparación a las herramientas actuales.

Referencias

[1] Gallo Beatriz P., Rivetti Esteban A., Estudio comparativo de desempeño de herramientas para el análisis forense de correos electrónicos.

[2] Manual de Laravel: https://legacy.gitbook.com/book/richos/laravel-5/details.

[3] Introducción a Apache Jena: http://jena.apache.org/tutorials/rdf_api.html.

[4] Introducción a Apache Fuseki: http://juanfelipe.info/node/181

[5] Gallo Beatriz P. de, Vegetti Marcela, Leone Horacio, *Hacía una Ontología para el soporte de la trazabilidad del correo electrónico en la Forensia Digital*, CIIDDI 2017, Cuba.

[6] Google Spreadsheets: https://www.google.com/intl/es/sheets/about/

[7] Manual de ExportSheetData: https://www.addictivetips.com/web/how-to-export-google-sheets-data-to-json-and-xml/

[8] Faker: http://www.conasa.es/blog/generando-datos-prueba-faker/.