ISBN 978-987-4417-90-9

Decisiones y lecciones aprendidas en un proceso ETL aplicado a sistemas con testimonios de delitos de lesa humanidad

David Troncoso[1] *, Agustina Buccella[1], and Alejandra Cechich[1]

GIISCO Research Group
Departamento de Ingeniería de Sistemas - Facultad de Informática
Universidad Nacional del Comahue
Neuquen, Argentina
david.troncoso, agustina.buccella, alejandra.cechich@fi.uncoma.edu.ar

Resumen El proceso de selección de las fuentes de datos (extracción-E), el procesado y adaptación de los datos (transformación-T) y la carga de los mismos a un repositorio (carga-L), recibe el nombre de *proceso ETL*. El diseño e implementación de estos procesos son áreas de estudio importantes debido a la proliferación de sistemas similares cuya información necesita ser integrada y reestructurada para ser de utilidad. En este trabajo describimos la ejecución de un proceso ETL aplicado a dos sistemas que almacenan declaraciones y testimonios de delitos de lesa humanidad. Siguiendo dos objetivos necesarios para la integración y explotación de la información, describimos decisiones realizadas y lecciones aprendidas.

Keywords: Proceso ETL, Análisis de Datos, Declaraciones y testimonios

1. Introducción

Para realizar un desarrollo orientado al análisis de los datos en cualquier organización, se debe primero seleccionar la información fuente que pueda resultar útil, procesarla para que tenga sentido, y luego almacenarla en algún tipo de repositorio, en la forma de depósitos de datos [6] o lagos de datos [3]. La característica principal de estos repositorios es que actúan como centros de información en donde se vuelcan, en algún formato específico, los datos de todas las fuentes que se deseen explotar en un proceso de extracción, transformación y carga (ETL)[2,4,5,7].

Para el diseño de procesos ETL existen en la literatura muchas propuestas diferentes las cuales podemos clasificar entre las que se basan en UML, en BPMN, y en el uso de ontologías [1]. Todas estas propuestas van dirigidas a simplificar y organizar el proceso ETL debido a la complejidad derivada de la

* Este trabajo esta parcialmente soportado por el Proyecto Desarrollo de Software basado en Reuso Parte II

ISBN 978-987-4417-90-9

forma en que pueden estar representadas las fuentes de información. Es muy común encontrarnos con información no estructurada, incompleta, inconsistente y/o redundante.

En este artículo presentamos una experiencia en el desarrollo de un proceso ETL aplicado sobre dos sistemas informáticos que recaban información de testimonios y juicios realizados por delitos de lesa humanidad. El primero, denominado *Sistema Informático de Procesamiento de Declaraciones en Juicios de DDHH* (SIPDJ) desarrollado por la Facultad de Informática (FAI) de la Universidad Nacional del Comahue (UNCO), gestiona toda la información generada en las declaraciones durante los juicios de la Escuelita I y II. Por otro lado el *Sistema de Análisis Sociológico de Querellas* desarrollado por el Equipo de Asistencia Sociológica a las Querellas (EASQ) a través del Centro de Estudios sobre el Genocidio (CEG)[1], comprende la Asistencia Sociológica a las Querellas (ASQ) almacenando datos correspondientes a parte del Juicio ABO (Atlético, Banco y Olimpo), Juicio Operativo Independencia de Tucumán y la mega causa de Santiago del Estero. Ambos sistemas gestionan información similar. La diferencia fundamental es que SIPDJ se centra en los juicios de DDHH y las declaraciones que se almacenan son las brindadas en los casos pertenecientes a dichos juicios. En cambio, ASQ se enfoca específicamente en registrar información necesaria para colaborar cualitativamente con las querellas de los juicios.

Como estos sistemas requieren almacenar información textual (de las declaraciones y testimonios) poseen atributos que tienen formatos de texto extensos y sin estructura. A su vez, como las declaraciones y/o testimonios no poseen información precisa, por ejemplo, el declarante no puede especificar lugares precisos donde estuvo o personas que lo retuvieron, hace más difícil la extracción de información para su análisis. La información está desnormalizada y es muchas veces redundante[2]. Con esto en mente, se definieron varios objetivos de manera de crear un depósito de datos consistente que permita un análisis de datos posterior. Entre esos análisis, en este trabajo nos centramos en el proceso ETL para dos objetivos específicos: (1) *Identificar a las personas que se nombraron en los testimonios para definir el recorrido a través de los centros clandestinos de detención (CCD)* y (2) *Obtener información georeferencial de las fuentes, para ser reflejada en un mapa interactivo. De esta forma se habilita la posibilidad de realizar análisis visuales respecto a la distribución geográfica de los datos.*

De esta forma este trabajo contribuye a mostrar una experiencia real de un proceso ETL sobre datos de texto, aportando decisiones y lecciones que pueden ser de interés en situaciones similares. El articulo se organiza de la siguiente manera. En la sección siguiente describimos el proceso realizado para cumplir con los dos objetivos propuestos. Luego en la Sección 3 analizamos las lecciones aprendidas para que sean útiles para la aplicación de procesos ETL con características similares. Finalmente se describen las conclusiones y trabajos futuros.

[1] `https://www.untref.edu.ar/instituto/ceg-centro-de-estudios-sobre-genocidio`
[2] Es importante aclarar que esta redundancia muchas veces es necesaria ya que intenta no obviar información útil de los testimonios y/o declaraciones

ISBN 978-987-4417-90-9

2. Aplicación de un Proceso ETL para normalizar e integrar información sobre sistemas de delitos de lesa humanidad

Como se mencionó previamente el proceso ETL debe estar orientado a preparar la información para cumplir con los dos objetivos propuestos. Para analizar cada objetivo y el proceso ETL definido, analizaremos el problema, su diseño y su implementación final o solución. Para el diseño utilizamos el enfoque definido en [5] el cual se basa en el uso de clases UML estereotipadas y mecanismos asociados.

2.1. Objetivo 1: Identificar las personas que se nombraron en los testimonios para definir el recorrido a través de los centros clandestinos de detención.

Tomamos como base las tablas *cumpa* y *testimoniante* del sistema ASQ, y *victima*, *persona*, *secuestro*, *traslado* y *cautiverio* del sistema SIPDJ. Por razones de espacio describimos algunas de las reglas más importantes aplicadas a la creación de una tabla *recorrido*. Sin embargo para cumplir el objetivo realizamos un cambio sobre la estructura de varias de las tablas mencionadas previamente. Para mostrar el panorama sobre la forma en que los datos están almacenados en la mayoría de estas tablas fuentes, en la Figura 1[3] mostramos algunos datos de la tabla *cumpa* del sistema ASQ. Esta tabla representa a los individuos secuestrados a quienes el testimoniante vio en los diversos CCD por los que pasó, y compañeros de quienes supo que allí estaban o habían estado[4].

Problemas encontrados

(1) *Valores nulos, repetidos.* En la figura podemos observar que hay algunos valores nulos sobre los atributos como *apellido1* y *apellido2*, *apodo*, etc. A su vez, algunos valores en los atributos se pueden repetir si en distintos testimonios se nombra a la misma persona o si el testimoniante se describe en diferentes CCD, lo que significa que por cada fila se repiten varios datos como *oriundo*, *ocupacion*, *sexo*, etc.

(2) *Información que puede resultar inconsistente, redundante e incluso incompleta.* También se da el caso que una persona fuese divisada por 2 o más testimoniantes, entonces aparecerá en la tabla con posibles datos repetidos con algunos detalles descriptos por cada testimoniante. De esta forma, la información puede resultar inconsistente, redundante e incluso incompleta. Sin embargo, a los principios para lo que fue creado el sistema, esto es útil ya que confirma la presencia de cierta persona en un centro clandestino, y se diferencian los datos aportados por cada testimoniante.

[3] Hemos difuminado algunos datos debido a la sensibilidad de los mismos.
[4] Esta tabla posee más de 30 atributos

ISBN 978-987-4417-90-9

Figura 1: Formas en que los datos se encuentran almacenados en la tabla *cumpa*

Diseño ETL. En la Figura 2 podemos observar la parte del diseño del proceso ETL desde el sistema ASQ que involucra las tablas *cumpa* y *testimoniante* para crear la tabla *recorrido*. Las decisiones involucran el uso de join, agregación y filtros.

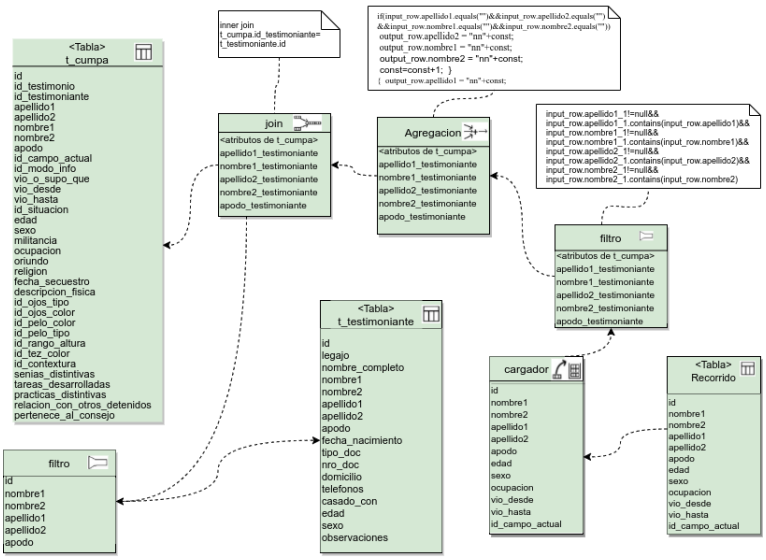

Figura 2: Parte del diseño ETL para crear la tabla *recorrido*

Componente de *Join*. La tabla *cumpa* cuenta con los atributos *nombre1*, *nombre2*, *apellido1*, *apellido2* y *apodo* que corresponden a la persona a la que se logra reconocer en un CCD, pudiendo ser una auto descripción del mismo testimoniante. Para detectar estos casos se busca, a través de este componente, dejar sobre cada registro de la tabla, también los atributos de *nombres*, *ape-*

ISBN 978-987-4417-90-9

llidos y *apodo* pero correspondientes al testimoniante (*nombre1_testimoniante*, *nombre2_testimoniante*, etc.).

Componente de *agregación*. Se encarga de asignar a cada uno de los registros que no tengan datos en ninguno de los atributos *nombre1*, *nombre2*, *apellido1*, *apellido2* un valor único. La finalidad de esto es que permanezca el resto de la información asociada a una persona de la cual no se supo nombre ni apellido.

Componente de *filtro*. Se coloca para apartar todos aquellos registros donde la persona que se describe es la misma que el testimoniante. Esto se realiza comparando los atributos *nombres1*, *nombre2*, *apellido1*, *apellido2* con *nombre1_testimoniante*, *nombre2_testimoniante*, *apellido2_testimoniante*, *apellido2_testimoniante*. El conjunto de datos que se corresponden con la condición son dispuestos sobre un componente *cargador* con los atributos seleccionados para ser ingresados a la nueva tabla *recorrido*.

Implementación de las decisiones. Para implementar el proceso ETL visto en la Figura 2 se utilizó la herramienta Talend Open Studio[5] la cual es una herramienta de código abierto y software libre. En la Figura 3 podemos observar la forma en que cada uno de los filtros, agregaciones y joins fueron implementados.

Al igual de como se detalló en el proceso de ETL, se parte de las 2 tablas (*cumpa* y *testimoniante*), para filtrar los atributos deseados del testimoniante mediante el elemento $<< tFilterColumns\_2 >>$. Luego, el componente $<< tMap\_1 >>$ realiza el *join* por *id_testimoniante* agregando a cada registro de *cumpa* los atributos detallado en el diseño (*nombre1_testimoniante*, *nombre2_testimoniante*, etc.) El componente $<< tJavaRow\_1 >>$ permite exponer cada atributo de cada fila del conjunto a un código en java. Con esto se realiza un chequeo de los valores de los atributos *nombre1*, *nombre2*, *apellido1* y *apellido2* en *cumpa*. Si todos ellos son nulos, se les cambia el valor por las siglas $<< nn >>$ concatenado a un valor incremental. Los elementos denominados CheckPoint corresponden a un componente «tFileOutputExcel» de Talend que permite volcar los valores en un XLS con fines de validar las operaciones sobre los datos y detectar errores. El elemento $<< tFilterRow\_1 >>$ se encarga de realizar el filtrado de aquellos registros donde *nombre1_testimoniante*, *nombre2_testimoniante*, *apellido1_testimoniante*, *apellido2_testimoniante* se correspondan con *nombre1*, *nombre2*, *apellido1* y *apellido2*. Este conjunto resultante del filtrado es verificado nuevamente para almacenarlo en la nueva tabla *recorrido*.

2.2. Objetivo 2: Obtener información georeferencial de las fuentes, para ser reflejada en un mapa interactivo.

Aquí, utilizamos la tabla *secuestros* de ambos sistemas y *victima*, *persona*, *traslado* y *cautiverio* del sistema SIPDJ. Para el caso de las tablas *secuestro* la información almacenada en ambos sistemas es similar con algunas diferencias en cuanto a datos que se pudieron recuperar de los testimonios, como la existencia

[5] `https://es.talend.com/products/talend-open-studio/`

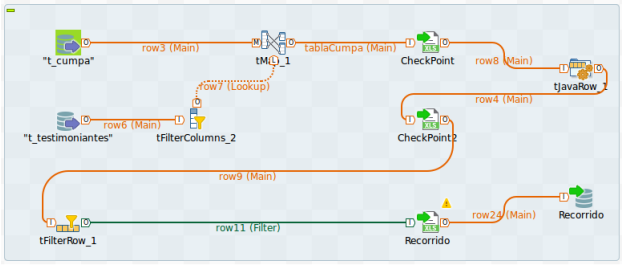

Figura 3: Fragmento del trabajo en Talend para crear la tabla *recorrido*

de otras personas secuestradas en el mismo periodo.

Problemas encontrados.

Información incompleta, formatos diferentes. En este caso, como queremos reconocer los lugares para georeferenciarlos en un mapa, analizamos las direcciones de los lugares del secuestro (o domicilios) tal como están almacenados. Como en el caso anterior, vemos en la Figura 4 una de las tablas *secuestros* donde las direcciones se encuentran incompletas y almacenadas en muchas formas diferentes. Por ejemplo se observan direcciones del tipo:

1. San Martín 151, San Miguel de Tucumán
2. Su domicilio, sito en el Pasaje Ecuador 135, barrio El Palomar en La Banda del Río Salí (Cruz Alta, Tucumán)
3. Su domicilio calle Uruguay 4532, San Miguel de Tucumán, Tucumán
4. Me secuestraron el 1ro. de Junio de 1978 en un bar a una cuadra de General Paz y Av. de los Constituyentes.

		lugar_secuestro	momento_del_dia_secuestro
1	5	lugar	momento
2	13	su domicilio	noche
3	1	"El día 2 de octubre de 1978 lo secuestran en Juan B.Alberdi 5045, lugar donde trabajaba en la ciudad de Buenos A	A las cinco de la tarde
4	40	su domicilio	en la madrugada
5	45	"A mí secuestran el 2 de octubre de 1978 cuando salía de mi trabajo en Juan Bautista Alberdi 5045 de esta ciudad.	a las 15.30 de la tarde
6	47	En la casa de su tía Carmen Aguiar de Lapacó, en Marcelo T. de Alvear 934, entre Carlos Pellegrini y Suipacha.	Noche, después de cenar
7	50	Domicilio particular (Lules, Tucumán)	aproximadamente a las dos de la
8	51	El Empalme, Ranchillos (Cruz Alta, Tucumán). ATENCIÓN: no lo explicita en el testimonio pero se infiere que fue allí	
9	52	San Miguel de Tucumán, Tucumán. ATENCIÓN: se infiere, no es especificado en el testimonio.	
10	52	San Miguel de Tucumán (Tucumán). ATENCIÓN: no lo dice el testimonio pero se infiere.	
11	54	Vía pública. La Cocha (La Cocha, Tucumán)	"fue detenido, mejor dicho secue
12	55	sin datos, [aparentemente] fue en Ingenio Lules (Lules, Tucumán)	
13	56	su domicilio, Pasaje Vyetes 1482, frente a la plaza del Barrio Victoria a la altura de la Avda. Alem 1450 (San Miguel d	por la mañana, tipo 7.00 hs

Figura 4: Formas en que el *lugar_secuestro* esta almacenado en la tabla *secuestro* de ASQ

Diseño ETL. El diseño ETL para analizar y mejorar las direcciones se puede observar en Figura 5. Las decisiones involucran el uso de filtros y agregación.

Uso de *filtros.* Aquí se decidió en primera instancia, una vez obtenidas las tablas de *secuestro* de ambos sistemas, unificar los mismos datos en una única tabla

ISBN 978-987-4417-90-9

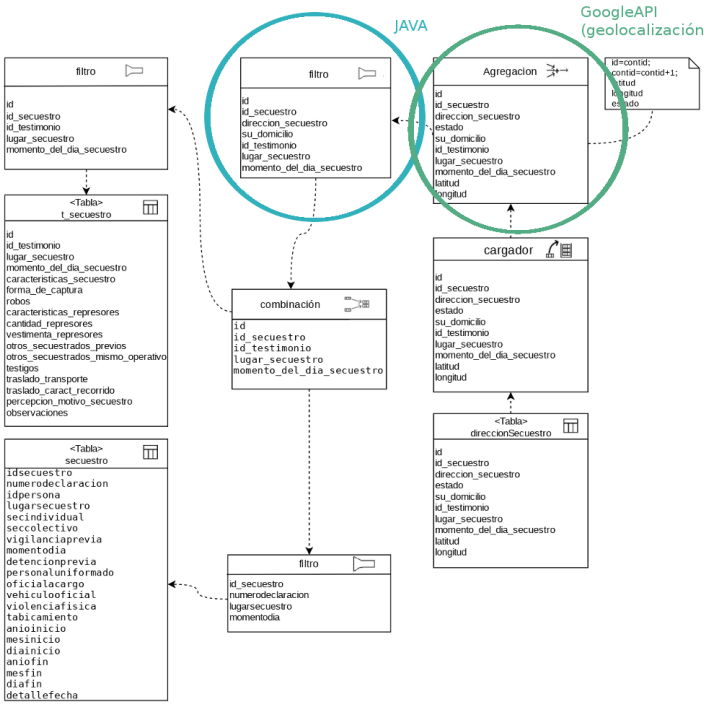

Figura 5: Diseño del proceso ETL para el segundo objetivo

llamada *direccionSecuentro*, con la información referente a la dirección de secuestro y algunos atributos mas. Así para ambas tablas se les aplica un componente *filtro* donde se dejan los atributos *id*, *id_secuestro*, *id_testimonio*, *lugar_secuestro* y *momento_del_dia_secuestro* para la tabla *secuestro* de ASQ y los atributos *id_secuestro*, *numerodeclaracion*, *lugarsecuestro* y *momentodia* de SIPDJ. Estos dos conjuntos se combinan en un nuevo *filtro* (en celeste en la Figura 5) encargado de conservar solo los datos que representan una dirección geográfica. Estos datos quedan almacenados en el nuevo atributo *direccion_secuestro*, y sobre otro nuevo atributo denominado *su_domicilio* se mantiene un valor que indica si sobre la información filtrada se nombró el domicilio de la víctima.

Componente de *agregación* (en verde en la Figura 5). Se presentan los datos filtrados a un sistema de geolocalización que busca convertir esas direcciones en coordenadas geográficas. Los datos que obtienen un resultado en un mapa son almacenados en los nuevos atributos *latitud* y *longitud*. Todos estos nuevos datos resultantes se preparan para ser almacenados mediante el componente cargador sobre la nueva tabla resultante *direccionSecuestro*.

Implementación de la solución. Aquí también se utilizaron algunas de las funcionalidades de Talend Studio, como por ejemplo filtrar y combinar ambas

ISBN 978-987-4417-90-9

tablas de *secuestro*. Luego, este nuevo conjunto, fue volcado sobre un archivo como entrada del algoritmo de filtrado desarrollado en java[6] y que posee como salida otro archivo normalizado (filtro en celeste en la Figura 5). A partir del filtrado y análisis de estas expresiones, se obtuvo que aproximadamente 162 registros contenían las expresiones «su domicilio» o «su casa», lo que representa un 44 % del total de registros en las tablas *secuestro*.

Luego definimos una repetitiva que analiza el archivo de texto como entrada y organiza los datos de salida acorde a los datos que espera un sistema de geo-referenciacion. Adicionalmente, el algoritmo crea un nuevo atributo numérico con valores del 1 al 4. Estos valores se generan de acuerdo a la precisión que podríamos obtener en el proceso de geo-referenciacion. El valor 1 representa que se disponen con todos los datos para que sea una ubicación específica en un mapa, en cambio, para los casos donde se tienen por ejemplo solo una calle con su altura y una provincia. Así este valor de define de la siguiente forma:

$$\left.\begin{array}{c} calle + altura + ciudad + provincia = 1 \\ calle + altura + [ciudad]|[provincia] = 2 \\ calle + altura = 3 \\ \emptyset = 4 \end{array}\right\} precision\_de\_direccion$$

Una vez finalizado el algoritmo, la dirección es procesada por la API de Google ($<< googleMapsClientgeocode >>$). Aquellos registros cuyo valor fue 4 no se exponen al siguiente proceso de detección, ya que se sabe que no van a poder ser geo-greferenciados. A partir del análisis con la API se lograron obtener aproximadamente 300 resultados geo-referenciables que se diferencian por el peso obtenido. Para finalizar se depositan los valores geo-referenciados sobre los atributos *latitud* y *longitud* de la tabla *direccionSecuestro* y se vuelcan en un archivo CSV, mediante Talend. Estos datos nuevos son visibles a través de una Web que se desarrolló para la consulta y visualización de los datos. En la Figura 6[7] podemos observar una pantalla del sistema donde al interactuar con un indicador (punto de geolocalización), se despliega un cuadro de información con un número que corresponde al *Id del registro*, y en la parte inferior la dirección geo-referenciada. A su vez podemos observar los diferentes colores mostrando gráficamente la precisión de la dirección mostrada. Las banderas que se pueden observar en el mapa representan los CCD, que se obtienen a partir de otras fuentes externas a las bases con las que se contaban, y son filtradas, geolocalizadas y almacenadas.

3. Lecciones Aprendidas

Luego de haber realizado el proceso ETL, podemos destacar las siguientes lecciones aprendidas:

[6] Aquí se decidió no utilizar la funcionalidad de Talend para expresiones regulares ya que es compleja y posee pocas formas de personalización

[7] Los datos mostrados no representan necesariamente la realidad

ISBN 978-987-4417-90-9

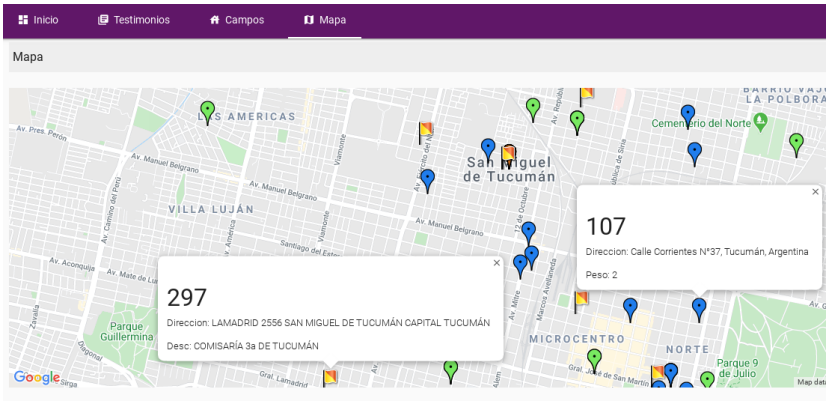

Figura 6: Interfaz del sistema mostrando indicadores geolocalizados

- *El conocimiento del dominio fue fundamental en el proceso*: el trabajo conjunto entre informáticos y expertos del dominio fue fundamental para interiorizarse sobre la forma en que la información estaba almacenada y la forma correcta de normalizarla o reestructurarla. Por la naturaleza del sistema, se almacenan muchos atributos de texto muy largos y con muchas descripciones. Estas descripciones a pesar de que a simple vista parecían redundantes, los expertos hicieron notar la importancia de tenerlas en las declaraciones o testimonios ya que recaban información no precisa, pero si importante sobre las detenciones. Por lo tanto, se debía realizar una extracción muy cuidadosa de los datos de forma tal de no eliminar dicha información. Es por esto que, por ejemplo, para el caso de los lugares de secuestro, se eligió utilizar colores para indicar la precisión de las direcciones.
- *La aplicación de una metodología de diseño del proceso ETL sirvió para organizar al equipo y reproducir prácticas*: el enfoque de diseño de ETL utilizado basado en UML también facilitó el trabajo debido a que UML era conocido y aplicado por los informáticos que participaron; sólo tuvieron que aprender los mecanismos particulares de ETL. Al mismo tiempo, la metodología permitió que se puedan replicar decisiones de diseño entre los problemas abordados; particularmente cuando se debieron cargar los CCDs en el mapa en donde muchas direcciones estaban también incompletas y se repitieron mecanismos realizados para el primer objetivo.
- *La aplicación de la herramienta open source Talend Studio agilizó también la implementación del proceso*: esta herramienta es muy intuitiva y sencilla, con lo que la implementación pudo realizarse relativamente rápido. A su vez, al tener el diseño ETL ya planteado, el mapeo entre éste y su implementación en Talend fue muchas veces trivial. Igualmente, en algunos casos tuvimos que agregar cierta programación para plasmar justamente las particularidades de este dominio.
- *El resultado del diseño e implementación del proceso ETL generó información útil para ser analizada*: aunque lo definimos fuera del alcance de este

trabajo, la información generada se encuentra en un formato ideal para ser analizada. Dentro de los dos objetivos planteados, se pueden efectuar análisis sencillos que involucren recorridos de personas por CCDs, cantidad de CCDs por los que pasó la misma persona, lugares donde fueron secuestrados y recorridos que realizaban una vez detenidos, etc. A su vez se podrían realizar análisis más complejos que ayuden a determinar patrones de comportamiento en cuanto a los secuestros, recorridos recurrentes sobre los CCDs, etc. Esto se plantea como trabajo futuro.

4. Conclusiones y Trabajo Futuro

En este trabajo hemos descripto el diseño e implementación de un proceso ETL aplicado a dos sistemas que almacenan información sobre declaraciones y testimonios de delitos de lesa humanidad. En particular nos hemos centrado en dos objetivos principales y en base a ellos hemos descripto el proceso realizado destacando las lecciones aprendidas durante el mismo.

Como trabajo futuro se plantea continuar registrando las decisiones de diseño e implementación derivadas de nuevos ETLs para ese dominio y a su vez realizar las tareas de análisis de datos propuestas para evaluar la real utilidad de la información resultante para efectuar análisis de datos complejos.

Referencias

1. Hamed, I., Ghozzi, F.: A knowledge-based approach for quality-aware etl process. In: 2015 6th International Conference on Information Systems and Economic Intelligence (SIIE). pp. 104–112 (2015)
2. Luján-Mora, S., Vassiliadis, P., Trujillo, J.: Data mapping diagrams for data warehouse design with uml. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.W. (eds.) Conceptual Modeling – ER 2004. pp. 191–204. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
3. Quix, C., Hai, R.: Data Lake, pp. 1–8. Springer International Publishing, Cham (2018), https://doi.org/10.1007/978-3-319-63962-8_7-1
4. Simitsis, A., Skoutas, D., Castellanos, M.: Natural language reporting for etl processes. In: Proceedings of the ACM 11th International Workshop on Data Warehousing and OLAP. p. 65–72. DOLAP '08, Association for Computing Machinery, New York, NY, USA (2008), https://doi.org/10.1145/1458432.1458444
5. Trujillo, J., Luján-Mora, S.: A uml based approach for modeling etl processes in data warehouses. In: Song, I.Y., Liddle, S.W., Ling, T.W., Scheuermann, P. (eds.) Conceptual Modeling - ER 2003. pp. 307–320. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
6. Vaisman, A., Zimnyi, E.: Data Warehouse Systems: Design and Implementation. Springer Publishing Company, Incorporated, 1st edn. (2016)
7. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual modeling for etl processes. In: Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP. p. 14–21. DOLAP '02, Association for Computing Machinery, New York, NY, USA (2002), https://doi.org/10.1145/583890.583893