Revista en Telecomunicaciones e Informática, Vol. 3, No. 5 p. 27 - 38
Medellín - Colombia. Enero - Junio de 2013, ISSN 2215-8200

Evaluación de la eficacia de la detección de duplicados utilizando Sql Server

Juan Pablo Pérez, Iván Amón

Resumen
Bajo el nombre de *Record Linkage* se conoce al conflicto que se presenta en los datos cuando una misma entidad del mundo real aparece representada dos o más veces a través de una o varias bases de datos, en registros o tuplas con igual estructura pero sin un identificador único y presentan diferencias en sus valores. En este artículo nos referiremos a esta problemática como detección de duplicados.

Para la detección de duplicados existen múltiples herramientas que utilizan funciones de similitud en la realización de esta tarea. Es bien conocido que algunas funciones de similitud son más eficaces que otras dependiendo de la situación problemática que presenten los datos. Por ejemplo q-grams realiza una mejor tarea de detección que la distancia de edición cuando se está en presencia de palabras en diferente orden. Asimismo, las diferentes herramientas pueden lograr diferentes grados de eficacia en la detección de duplicados dependiendo de varios factores. En este artículo se presentan los resultados de una evaluación realizada a la herramienta para detección de duplicados *Fuzzy Lookup* que viene incluida en el SSIS (*Sql Server Integration Services*) de Microsoft Sql Server 2008 R2. Los resultados muestran que su eficacia es variable dependiendo de la situación problemática que presenten los datos.

Palabras Clave: Calidad de Datos, Detección de Duplicados, *Fuzzy Lookup, Record Linkage.*

I. Introducción

En los datos empresariales se presenta con relativa frecuencia que una misma entidad se encuentra almacenada en forma no idéntica más de una vez. El hecho de que esté almacenada en forma similar más no idéntica, dificulta la tarea de detección y depuración de estos casos requiriéndose de algoritmos especializados para lograr identificarlos. Para entender en detalle la problemática de la detección de duplicados véase [1]. En [2] los autores presentan varias "situaciones problemáticas" que ilustran las diferentes variaciones que pueden tener los datos haciendo que un mismo objeto no esté representado en forma idéntica y las usan para comparar la eficacia de diferentes funciones de similitud. La tabla 1 presenta estas situaciones problemáticas.

Tabla 1. Situaciones problemáticas

Situación Problemática	Ejemplo
Errores ortográficos (ERR)	"Jorge Eduardo Rodríguez López" vs. "Jorje Eduadro Rodrígues Lopes"
Abreviaturas: truncamiento de uno o más *tokens* (ABR)	"Jorge Eduardo Rodríguez López" vs. "Jorge E Rodríguez L"
Tokens faltantes: eliminación de uno o más tokens (TFL)	"Jorge Eduardo Rodríguez López" vs. "Jorge Rodríguez"
Prefijos/sufijos sin valor semántico: presencia de subcadenas al principio y/o al final (PSF)	"Jorge Eduardo Rodríguez López" vs. "PhD Jorge Eduardo Rodríguez López, UPB"
Tokens en desorden (TDR)	"Jorge Eduardo Rodríguez López" vs. "Rodríguez López Jorge Eduardo"
En el resto del artículo se hará mención de las situaciones problemáticas con los términos abreviados (ERR,ABR, TFL,PSF y TDR).	

Las funciones de similitud sirven para detectar qué tan similares son dos contenidos y se materializan en algoritmos de computación que realizan esta tarea. Con el paso de los años, los investigadores han desarrollado múltiples funciones de similitud y está ampliamente documentado que no todas son igual de eficaces ante la presencia de datos con las diferentes situaciones problemáticas [1].

Las herramientas de software existentes hacen uso de estas funciones de similitud, bien sea en forma trasparente o visible para el usuario. Esto es, las primeras no

le solicitan al usuario que seleccione la función de similitud a usar en la tarea de detección de duplicados, sino que ellas se encargan en forma transparente para el usuario de utilizar la(s) que considere la herramienta. Las segundas permiten que el usuario indique las funciones a utilizar, lo cual obviamente requiere algún conocimiento sobre su funcionamiento por parte del usuario. Este factor, unido a otros, puede influir en la calidad de la detección de duplicados que pueda lograrse con una u otra herramienta, siendo importante poder establecer la eficacia de éstas.

Los trabajos investigativos que evalúan la eficacia de las herramientas para detección de duplicados son pocos. En [3] los autores proporcionan una visión global de los algoritmos para la detección de duplicados y de las herramientas para este propósito y realizan una comparación entre algunos sistemas comerciales que aplican técnicas para detección de duplicados, como *Oracle Warehouse Builder 10gR2*, *IBM's Entity Analytic Solutions y SQL Server Integration Services 2005,* este último aplicando las tareas *Fuzzy Lookup* y *Fuzzy Grouping*. Las conclusiones fueron que algunos de estos sistemas tienen ventajas sobre los otros ya que unos soportan varias funciones de similitud y tienen diferentes metodologías para realizar la tarea de *Record Linkage.*

De otra parte, en el año 2010 para el censo en Brasil se realizaron algunas mejoras en cuanto a incorporación de nuevas metodologías y tecnologías. El desarrollo de paquetes de software o herramientas que implementan modelos computacionales para la detección de duplicados se había incrementado en los últimos años, y era posible encontrar varios programas que realizaban tareas similares. Para poder elegir el mejor software para la detección de duplicados del censo de Brasil, los investigadores da Silva y otros [4] del IBGE (Instituto Brasileño de Geografía y Estadística) evaluaron algunas características de cada software como aspectos operativos, tipos de licencia, metodologías que utilizaban, entre otras, y así elegir la herramienta adecuada. Se concluyó que Relais [5] y Febrl [6] eran las herramientas más completas de las evaluadas y específicamente Relais cubría mejor las necesidades para las necesidades del censo. Sin embargo para el proyecto esto no era suficiente y fue necesario desarrollar un software en lenguaje R.

Microsoft ofrece la funcionalidad de detección de duplicados en su producto Sql Server 2008 R2 a través de la herramienta *Fuzzy Lookup* [7] que es un componente del SSIS. De nuevo es muy poco lo que puede encontrarse a nivel investigativo acerca de esta herramienta. *Microsoft Research* [8], a través de Arvinda y otros [9], se limitan a presentar las herramientas *Fuzzy Lookup* y *Segmentation* indicando cómo han sido aplicadas en situaciones reales como las personas desaparecidas en el huracán Katrina, búsquedas de monumentos y ubicaciones por el servicio de *bing* y para los registros coincidentes de todos sus clientes, entre otras.

Ya que esta herramienta es una de las que utiliza las funciones de similitud en forma transparente para el usuario (no permite al usuario seleccionar la técnica para detección de duplicados sino que usa algunas técnicas no explícitas para realizar la

labor de detección), es interesante evaluar qué tan eficaz es su tarea de detección de duplicados ante las diferentes situaciones problemáticas. Al no encontrarse evidencia de trabajos que evalúen la eficacia de esta herramienta y mucho menos bajo diferentes situaciones problemáticas, el aporte de este artículo consiste en proveer a los usuarios interesados en utilizar herramientas de *Record Linkage* criterios para su uso.

El resto de este artículo está distribuido así: la sección 2 presenta algunas de las funciones de similitud comúnmente ofrecidas por las herramientas para detección de duplicados, la sección 3 presenta información sobre *Fuzzy Lookup*, la sección 4 presenta la metodología utilizada para la evaluación de *Fuzzy Lookup*, la sección 5 presenta los resultados encontrados y por último se presentan las conclusiones.

II. Algunas funciones de similitud

A continuación se presentan algunas de las funciones de similitud ofrecidas por herramientas de detección de duplicados como TAILOR [10], Fril [11], Relais [5], Febrl [6], MTB - ToolBox (MTB) [12].

A. Distancia de edición

La distancia de edición entre dos cadenas de texto A y B se basa en el conjunto mínimo de operaciones de edición necesarias para transformar A en B (o viceversa). Las operaciones de edición permitidas son eliminación, inserción y substitución de un carácter. En el modelo original, cada operación de edición tiene costo unitario, siendo referido como distancia de Levenshtein [13].

B. Distancia de brecha afín

La distancia de edición y otras funciones de similitud tienden a fallar identificando cadenas equivalentes que han sido demasiado truncadas [1], ya sea mediante el uso de abreviaturas o la omisión de tokens ("Jorge Eduardo Rodríguez López" vs "Jorge E Rodríguez"). La distancia de brecha afín ofrece una solución al penalizar la inserción/eliminación de k caracteres consecutivos (brecha) con bajo costo, mediante una función afín $\rho\,(k) = g + h \cdot (k - 1)$, donde g es el costo de iniciar una brecha, h el costo de extenderla un carácter, y $h \ll g$ [14].

C. Similitud Smith-Waterman

La similitud Smith-Waterman entre dos cadenas A y B es la máxima similitud entre una pareja (A′, B′), sobre todas las posibles, tal que A′ es subcadena de A y B′ es subcadena de B. Tal problema se conoce como alineamiento local. El modelo original

de Smith y Waterman define las mismas operaciones de la distancia de edición, y además permite omitir cualquier número de caracteres al principio o final de ambas cadenas. Esto lo hace adecuado para identificar cadenas equivalentes con prefijos/ sufijos que, al no tener valor semántico, pueden ser descartados. Por ejemplo, "PhD Jorge Eduardo Rodríguez López" y "Jorge Eduardo Rodríguez López, Universidad Pontificia Bolivariana" tendrían una similitud cercana a uno [15].

D. Similitud de Jaro

Jaro desarrolló una función de similitud que define la trasposición de dos caracteres como la única operación de edición permitida. Los caracteres no necesitan ser adyacentes, sino que pueden estar alejados cierta distancia d que depende de la longitud de ambas cadenas [16].

E. Jaro-winkler

Es una variante de la similitud de Jaro. La distancia métrica Jaro-Winkler está diseñada y es más adecuada para cadenas cortas tales como nombres de personas. La puntuación se normaliza de forma que 0 equivale a ninguna similitud y 1 es una coincidencia exacta [17].

F. Similitud de q-grams

Un *q-gram*, también llamado *n-gram*, es una subcadena de longitud q. El principio tras esta función de similitud es que, cuando dos cadenas son muy similares, tienen muchos *q-grams* en común. Es común usar *bi-grams* o *di-grams* (q=2) y *tri-grams* (q=3) [18].

G. Soundex

Soundex es un algoritmo fonético para indexar textos por su sonido, al ser pronunciados en Inglés. El objetivo básico es codificar de la misma forma los textos con la misma pronunciación [19].

H. Double Metaphone

Metaphone es otro algoritmo fonético desarrollado por Lawrence Philips como respuesta a las deficiencias del algoritmo Soundex. Es más exacto que Soundex porque "entiende" las reglas básicas de pronunciación en inglés. El autor más tarde desarrolló una nueva versión del algoritmo, al que llamó "Double Metaphone", que produce resultados más exactos que el original. El algoritmo produce claves como salida. Palabras que suenen parecido comparten la misma clave y son de longitud variable [20].

III. Fuzzy Lookup – Sql Server 2008 Integration Services (Siss)

SSIS es una herramienta de Microsoft incluida en Sql Server 2008 que permite realizar tareas de estandarización y transformación de datos dentro de las cuales se encuentra la tarea *Fuzzy Lookup* que realiza funciones de detección de duplicados.

El procedimiento para realizar la operación de *Fuzzy Lookup* es el siguiente: se construye un flujo de datos con una fuente u origen de datos de entrada que puede ser de diferente tipo (excel, txt, tabla de base de datos, entre otras). Ésta se compara con una tabla de referencia donde están las posibles coincidencias y la herramienta devuelve los registros coincidentes en una tabla de salida con el formato deseado.

Fuzzy Lookup requiere tres parámetros para personalizar las búsquedas: número máximo de coincidencias por búsqueda, delimitadores de tokens y umbral de similitud. El número máximo de coincidencias hace referencia a la cantidad máxima de resultados similares a reportar por parte de la herramienta. Los delimitadores de tokens son los símbolos separadores de palabras o tokens. El umbral de similitud permite establecer a partir de qué grado de similitud se presenta un resultado como posible duplicado. Cuanto más cerca de 100%, será más alta la exigencia en cuanto a la similitud de los objetos.

Nótese cómo no es posible seleccionar la función de similitud a utilizar ni asignar porcentajes diferentes a los diferentes atributos que participan en la detección de duplicados como lo hacen otras herramientas como Fril [11].

IV. Diseño del experimento para la evaluación de la herramienta

Para la evaluación de la eficacia de la herramienta *Fuzzy Lookup* – SSIS 2008 R2 de Microsoft [7], se construyeron datos sintéticos usando el siguiente procedimiento:

Mediante la herramienta Web *FakeNameGenerator* (www. fakenamegenerator.com) se generaron conjuntos de datos compuestos por registros con campos como nombre, apellidos, dirección, ciudad, ocupación, teléfono, entre otros. En total se generaron diez conjuntos de datos cada uno con 500 tuplas o registros.

A partir de estos registros, se derivaron otros de acuerdo con la variación textual o situación problemática a probar (Ver Tabla 1). Así, para la situación problemática ABR (Abreviaturas), se generaron nuevas cadenas a las cuales se les recortaron los segundos nombres, segundos apellidos y las ocupaciones, dejando sólo la inicial o primera letra

de estos. Para la situación problemática TDR (Tokens en desorden), se cambió al nombre completo el orden de las palabras colocando primero los dos apellidos y luego los nombres. Para la situación problemática ERR (Errores ortográficos), se generaron nuevas cadenas a las cuales se les introdujeron errores de ortografía cambiando unas letras por otras (por ejemplo g por j, v por b, c por s, entre otras). Para la situación problemática PSF (prefijos/sufijos), se generaron nuevas cadenas a las cuales se antepuso o pospuso al nombre completo de la persona, textos de diferentes longitudes como Phd, Sra, Ingeniero, Universidad Pontificia Bolivariana, entre otros. Para la situación problemática TFL (Tokens Faltantes), se generaron nuevas cadenas a las cuales se les omitió una o varias palabras. En resumen, para cada uno de los diez conjuntos de datos originales de 500 registros, se derivaron otros diez conjuntos de datos con cinco registros por cada uno de los registros originales correspondientes a las cinco diferentes variaciones o situaciones problemáticas dando como resultado diez archivos con 2500 registros cada uno. La tabla 2 presenta dos ejemplos de los datos en los archivos originales y de los datos derivados con las variaciones.

Tabla 2. Archivos originales y archivos de datos derivados con variaciones

Dato archivo original	Dato derivado con las variaciones
Laurent Eitan Caballero Montalvo	Caballero Montalvo Laurent Eitan (TDR)
	Laurent E Caballero M (ABR)
	PhD Laurent Eitan Caballero Montalvo PhD (PSF)
	Lxurent Eitxn Cabxller Mntalvo (ERR)
	Laurent Caballero (TFL)
Gerald Albino Mireles Ochoa	Mireles Ochoa Gerald Albino (TDR)
	Gerald A Mireles O (ABR)
	Ingeniero Gerald Albino Mireles Ochoa (PSF)
	Gersld Qlbin Mireles Ocha (ERR)
	Gerald Mireles (TFL)

A continuación, usando la herramienta *Fuzzy Lookup*, se realizó la comparación entre los diez conjuntos de datos generados originalmente con los diez conjuntos de datos derivados para encontrar las posibles coincidencias. El objetivo es identificar qué tan eficaz es la herramienta en la detección de las cinco situaciones problemáticas planteadas. Esto se hizo con la siguiente configuración dentro de *Fuzzy Lookup*: Umbral de similitud: 70%, Número máximo de coincidencias por búsqueda: diez. Aunque existen algunos algoritmos que ayudan en la determinación del umbral adecuado y obtener así resultados óptimos [21], en este trabajo se fijó el umbral en forma empírica en 70% teniendo en la cuenta que si se asigna un valor muy alto cercano al 100%, aumenta la posibilidad de obtener falsos negativos (duplicados no reportados por la aplicación)

y si se fija muy bajo, aumenta la posibilidad de obtener falsos positivos (duplicados reportados por la aplicación que realmente no son duplicados).

Por último se analizaron los archivos de salida que genera la herramienta con los posibles duplicados y se determinó la cantidad de coincidencias encontradas para cada una de las variaciones o situaciones problemáticas. Nótese cómo en cada uno de los archivos originales existían 500 registros y en los archivos derivados había 500 registros más por cada una de las cinco variaciones o situaciones problemáticas planteadas y por tanto 500 casos es el número ideal de coincidencias que la herramienta podría encontrar para cada situación problemática.

V. Resultados

La tabla 3 presenta las coincidencias encontradas por la herramienta bajo la configuración mencionada, esto es, con un umbral de similitud del 70% y con un número máximo de diez coincidencias. La tabla 4 presenta la misma información pero en términos de porcentajes.

Como puede verse, para los datos propuestos y con la configuración suministrada, *Fuzzy Lookup* no detectó como coincidentes ningún caso de las abreviaturas propuestas (ABR) ni de los tokens faltantes (TFL). Para la situación de tokens en desorden (TDR) su eficacia fue demasiado baja, mejorando para los errores ortográficos y tipográficos y siendo altamente eficaz para los prefijos y sufijos.

Tabla 3. Resultados en cantidades de la comparación

Tipo Variación	ABR	TDR	ERR	PSF	TFL
Resultado 1	0	2	260	500	0
Resultado 2	0	1	244	500	0
Resultado 3	0	0	290	500	0
Resultado 4	0	0	260	500	0
Resultado 5	0	2	279	500	0
Resultado 6	0	1	283	500	0
Resultado 7	0	3	264	500	0
Resultado 8	0	4	253	500	0
Resultado 9	0	1	264	500	0
Resultado 10	0	1	255	500	0
Total	0	15	2,652	5,000	0

Tabla 4. Resultados en porcentaje de la comparación

Tipo Variación	ABR	TDR	ERR	PSF	TFL
Resultado 1	0	0.4	52.0	100	0
Resultado 2	0	0.2	48.8	100	0
Resultado 3	0	0.0	58.0	100	0
Resultado 4	0	0.0	52.0	100	0
Resultado 5	0	0.4	55.8	100	0
Resultado 6	0	0.2	56.6	100	0
Resultado 7	0	0.6	52.8	100	0
Resultado 8	0	0.8	50.6	100	0
Resultado 9	0	0.2	52.8	100	0
Resultado 10	0	0.2	51.0	100	0
Promedio	0	0.3	53.0	100	0

La tabla 5 resume los resultados de los falsos positivos y falsos negativos extraídos de las matrices de confusión o matrices 2x2 correspondientes a cada situación problemática.

Tabla 5. Resultados en una matriz de confusión

Tipo Variación	F-P	F-Neg	F-P(%)	F-Neg(%)
ABR	0	5,000	0	100
TDR	0	4,985	0	99.7
ERR	0	2,348	0	46.96
PSF	0	0	0	0
TFL	0	5,000	0	100
Esta tabla corresponde al total de todos los datos duplicados por cada situación problemática en los archivos derivados.				
F-P (Falsos positivos): Duplicados reportados por la aplicación que realmente no son duplicados. F-Neg (Falsos negativos): Duplicados en los datos originales que no fueron reportados por la aplicación.				

VI. Conclusiones

El experimento realizado permite concluir que para los datos de prueba y la configuración suministrada, *Fuzzy Lookup* no fue nada eficaz ante las variaciones o situaciones problemáticas de abreviaturas, tokens faltantes y tokens en desorden. Para el caso de variaciones por errores ortográficos y tipográficos logró detectar aproximadamente la mitad de los casos mientras que para la situación de prefijos y sufijos (sin importar la longitud de estos) su eficacia es total.

Recuérdese que esta herramienta no permite seleccionar la técnica para detección de duplicados (distancia de edición, *q-grams*, Jaro, etc) sino que usa algunas técnicas no explícitas para realizar la labor de detección. Como se indicó, algunas técnicas son más eficaces que otras dependiendo de la variación o situación problemática que presenten los datos. Si una herramienta no es capaz de identificar qué tipo de problemas tienen los datos específicos sobre los cuales va a trabajar (lo cual es muy difícil en forma automática), es posible que no utilice la técnica más adecuada para realizar una buena tarea de detección de duplicados para un conjunto de datos particular.

Teniendo en la cuenta lo anterior, los autores de este artículo consideran altamente deseable que una herramienta para detección de duplicados permita al usuario la selección de la técnica a aplicar. Sirva de ejemplo la siguiente situación: si el usuario que conoce los datos detecta mediante muestreo la existencia de muchos casos con los tokens en desorden (Juan Alberto López Gómez vs López Gómez Juan Alberto), puede indicar a la herramienta aplicar la técnica *q-grams* y no distancia de edición ya que está comprobado que para esta situación *q-grams* es más eficaz [2]. Obviamente indicar la técnica requiere usuarios más calificados que comprendan el funcionamiento, las fortalezas y debilidades de las técnicas, pero en las manos adecuadas pueden lograrse mejores resultados.

Como trabajos futuros se plantea realizar esta evaluación para otras herramientas como *Data Quality* de SAP, FRIL, Relais, entre otras. Asimismo, realizar un comparativo entre varias de ellas, examinando su eficacia ante las diferentes variaciones o situaciones problemáticas planteadas en este artículo.

Referencias

[1] A.K. Elmagarmid, P.G. Ipeirotis and V.S. Verykios, "Duplicate Record Detection; A Survey", *IEEE Transactions on Knowledge and Data Engineering*, vol.19, no.1, pp.1-16, Ene.2007.

[2] I. Amón,"Detección de duplicados: Una guía metodológica", *Revista colombiana de computación,* vol.11, no.2, pp.7-28, Dic.2010.

[3] *N. Koudas, S. Sarawagi, D. Srisvastava,"Record linkage: similarity measures and algorithms", in Int. Conf. on Management of Data, New York, 2006, pp.802-803.*

[4] A. Dinis da Silva, O. Santana, T. Silva, V. Layter," Study of Record Linkage Software for the 2010 Brazilian Census Post Enumeration Survey", Brazilian Institute of Geography and Statistics., Rio de Janeiro, CEP 20.031-170,Jul.2010.

[5] *Relais User's guide, v. 2.2*, European Commission, Unión Europea, 2009.

[6] *Febrl: Freely extensible biomedical record linkage, Release 0.1,* Australian National University, Canberra, 2002.

[7] Microsoft Corp. (2008) Fuzzy Lookup Transformation in SQL Server Integration Services 2008 Homepage on MSDN. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms137786(v=sql.100).aspx

[8] Wikipedia. (2013) Microsoft Research Homepage on Wikipedia. [Online]. Available: http://es.wikipedia.org/wiki/Microsoft_Research

[9] A. Arasu, S. Chaudhuri, Z. Chen, K. Ganjam, R. Kaushik, V. Narasayya,(2011).Towards a Domain Independent Platform for Data Cleaning. Microsoft Research. [Online]. Available FTP: ftp://ftp.research.microsoft.com/pub/debull/A11Sept/cleaning.pdf.

[10] M.G. Elfeky, V.S. Verylios, A.K. Elmagarmid, "TAILOR: A Record Linkage Toolbox," *in 18th Int. Conf. on Data Engineering*, Washington, 2002, pp.17.

[11] *FRIL: Finegrained Record Integration and Linkage Tool- Tutorial, v. 3.2*, Emory University, Atlanta, GA, 2009.

[12] *Merge ToolBox – MTB Record Linkage Software, v.0.74*, German Record Linkage Center, Regensburger, BY, 2012.

[13] V.I .Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals" *Soviet Physics Doklady*, vol.10, no.8, pp. 707-710, 1966.

[14] O. Gotoh, "An Improved Algorithm for Matching BiologicalSequences", *Journal of Molecular Biology*, vol. 162, no. 3, pp. 705-708, 1982.

[15] T.F.Smith, and M.S. Waterman,"Identification of Common Molecular Subsequences", *Journal of Molecular Biology*, vol. 147, no.1, pp. 195-197, 1981.

[16] M.A. *Jaro, Unimatch: A Record Linkage System: User's Manual*, US Bureau of the Census, Washington, D.C, 1976.

[17] W.E.Winkler, "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage", *Proceedings of the Section on Survey Research Methods*, pp. 354-359, 1990.

[18] W.E.Yancey,"Evaluating String Comparator Performance for Record Linkage", *Proceedings of the Fifth Australasian Conf. on Data mining and Analytics*,Sydney,2006,pp. 23-21.

[19] Wikipedia. (2013) Soundex Homepage on Wikipedia. [Online]. Available: http://es.wikipedia.org/wiki/Soundex

[20] Wikipdia. (2013) Double Metaphone Homepage on Wikipedia. [Online]. Available: http://es.wikipedia.org/wiki/Metaphone

[21] S. Chaudhuri, V. Ganti, R. Motwani, "Robust Identification of Fuzzy Duplicates", in *21st Int.Conf. on Data Engineering*, Washington, 2005, pp.865-876.

Autores

Juan Pablo PÉREZ RUIZ. Nació en Medellín, Antioquia en 1985. Terminó sus estudios de bachillerato en el año 2001 en el Liceo Municipal Concejo de Medellín y sus estudios de Ingeniería Informática en la Universidad Pontificia Bolivariana en el año 2012.

Iván AMÓN URIBE, MsC. Ingeniero de Sistemas y Especialista en Técnicas Computarizadas de Producción de la Universidad Eafit y Magíster en Ingeniería de Sistemas de la Universidad Nacional Sede Medellín. Más de 20 años de experiencia empresarial en cargos administrativos y de tecnología en diversas empresas y más de 20 años de experiencia como docente de pregrado y postgrado en diferentes universidades colombianas.

Experiencia práctica e investigativa en calidad de datos con ponencias y publicaciones internacionales y nacionales. Consultor empresarial en áreas de tecnología, especialmente en Gobernabilidad de Datos. Actualmente es docente de la facultad de ingeniería informática de la Universidad Pontificia Bolivariana y coordinador académico de la Especialización en Inteligencia de Negocios de esta Universidad.