Universidad de las Ciencias Informáticas

Facultad 3

Título: Implementación de algoritmos para la limpieza de datos

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autores: Evelyn Estoque Cabrera

Lianet Baró Galán

Tutor: Ing. Mailen Edith Escobar Pompa

Ciudad Habana

Junio, 2013

DECLARACIÓN DE AUTORÍA

Declaramos ser autores del presente trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _2013_.

Lianet Baró Galán

_____

Firma del Autor

Evelyn Estoque Cabrera

_____

Firma del Autor

Ing. Mailen Edith Escobar Pompa

_____

Firma del Tutor

DATOS DE CONTACTO

Tutor

Nombre y Apellidos: Ing. Mailen Edith Escobar Pompa.

Institución: Universidad de las Ciencias Informáticas (UCI).

E-mail: meescobar@uci.cu

Graduada de la Universidad de las Ciencias Informáticas en el año 2008.

Evelyn

Dedico este trabajo de diploma a mi madrecita Silvia por ser el motor impulsor y mi as guía durante todos mis años de estudio.

A mi papito Eulogio por ser mi ídolo adorado y el mejor padre del mundo.

Al solecito de mi vida, mi hermanita querida, por llenar mi vida de tanta alegría.

A mis abuelas y abuelos, porque aun sin tener un nivel elevado de profesión siempre supieron guiarnos por el camino del deber, en especial a mi abuela Pancha, a la cual me hubiera encantado mostrarle mi título de ingeniera.

A mi tía Sonia y Brígida por ser como mis madres en este tiempo y por su apoyo durante mi estancia en la universidad.

A mi novio Pedro, por ponerme mano dura y secar mis lágrimas cuando pensaba que no iba a poder.

A la Revolución por haber forjado universidades para crear jóvenes profesionales.

Lianet

A mis padres (Mi Mamita Linda y Mi Papito), porque sin su amor incondicional y apoyo no hubiese sido posible lograr mis metas.

A mis abuelos (Mima y Papú) por cuidarme en todas las etapas de mi vida.

A mi hermana Yudelvis por guiarme para ser buena profesional.

A mis primos Idalvis, Michel y Dini que siempre estuvieron allí para ayudarme a corregir todas mis fallas.

A mi familia por hacerse ver que soy parte de ella.

A la memoria de mis seres queridos en especial a la de mi hermano.

Al comandante Fidel por crear universidades del futuro.

Evelyn

Por estos 5 años en la universidad, por la vida, por los momentos más lindos y por no permitir que me rindiera jamás ante ninguna situación, le quiero agradecer a mi mamá por siempre orientarme hacia un futuro mejor y aguantar todo este tiempo lejos de mí. A mi papá por sus consejos y charlas educativas, por estar siempre ahí para mí. A mi hermana linda, porque si no existieras mi vida fuera vacía y sin mucha gracia. A mi tía Sonia, Susana, Soraya y Brígida por ser un ejemplo para mí de abnegación y sacrificio. A mis primos Ale, Carlos y Dyron por ser mis cómplices en cada travesura de mi linda infancia. A mi novio querido por estar a mi lado y soportar mis malacrianzas y caprichos. A toda la familia de mi novio que siempre me apoyo y se preocupó porque todo esto saliera bien. A Yadelkis mi amiga de andanzas por Camagüey, gracias por tu apoyo, a pesar de que te dejé 5 años solita, mi hermanita mayor. A mi compañera de tesis porque sin ella la culminación de este duro camino no se hubiese concretado. A ti Lianet por compartir conmigo estos 5 años que fueron en algún momento difíciles, no existen siquiera palabras para describir lo mucho que te debo en la vida, voy a extrañar mucho tu compañía. A Kirenia porque contigo a veces no sentía la ausencia de mi mamá a mi lado, fuiste esa persona mayor de buenos consejos y palabras de aliento. A mis amigas del politécnico Aimet, Elizabeth y Yarelis por ser parte de mi vida en esa etapa tan convulsa. A todo mi grupo desde que éramos 4101 hasta el 3502 por su apoyo en los momentos difíciles, en los que me vi en apuro por algún que otro examen. A Ariel Ahogando que aunque no llego con nosotros al final fue el mejor amigo que pude tener en la universidad. A mis compañeros del deporte, especialmente a Yoel y a Yadian por siempre confiar en mí. A mi tutora Mailen, porque en todo el transcurso de la tesis fue una tesista más apoyándonos y preocupándose porque no nos quedáramos atrás. Al profe Erich, Joisel, Adrián Naranjo, Fernando, Yusmara por su apoyo durante esta etapa. Quiero agradecer a todos aquellos que de una forma u otra hicieron posible que toda esta difícil batalla terminara con una gloriosa victoria.

Lianet

A mi Mamita Linda por su amor de madre infinito, por tomar la maravillosa decisión de traerme al mundo, por hacerme reír en cada conversación, por ser como es, una excelente madre. A mi Papito, por guiarme y apoyarme en mis trayectos estudiantiles, por darme propuestas entregándome siempre la oportunidad de escoger mí camino a transitar. A los dos, gracias, muchas gracias especialmente por hacerme ver que sigo siendo su niña chiquitica. A mis abuelos por aun consentirme y cuidarme. A mi hermana por soportarme, indicarme, por haberme dado ese sobrino chulo y lindo que es Oriel Jesús, para ti Nené de su

tía, besos. A mi prima Idalvis por ser como una hermana, ayudarme, cubrirme en momentos de fallas. A mis primos Michel y el loco del Diny, a Tita Martica, Tito Santiaguito y Tito Ernesto gracias por brindarme afecto en cada oportunidad.

A mi amigo John (Jonci) por compartir conmigo tragos dulces y amargos de la vida, por defenderme frente a todos sin importarle nada ni nadie. Espero que sigamos siendo amigos como hasta ahora.

A Adonys (Nenesito) que más que amigo, fue un gran novio, el hombre que toda mujer desearía tener para compartir toda una vida, nunca olvides que siempre estaré allí para ti, principalmente cuando estés bohémico y sé que me ayudaras y aconsejaras en todo momento.

A los integrantes del grupo folklórico Ilu Ashe que hicieron de mis cursos un lindo show, principalmente a mi amigo y profesor Herson que me formó como profesional del arte, además de enseñarme los primeros pasos de la lógica de Programación, aunque siempre les daba mis buenos aportes.

A mi grupo, en especial a Javier (Fashion), Yenisleidys (Yeni), Rayner (Raynito, La Bestia), Rolando, Meylin, Yoandy (Yoa, Yondi), Reinier. A mis amigos de la universidad Guillermo, Oniel, Hurshel, a todos cuídense y gracias por haber compartido un determinado momento junto a mí.

A Mi Chuqui por darme amor. A Deyler por contar con su amistad, por todas esas ocurrencias, pero siempre llevando en mente que conmigo hay que contar.

A mi amiga Kirenia por los regaños de todos los días, por tratarme como una hermana desde el momento que nos conocimos, por aconsejarme en determinadas situaciones donde tenía las respuestas equivocadas o inconclusas.

A mi compañera de tesis y mejor amiga desde primer año Evelyn, a ti Eve gracias por estar junto a mí en momentos que no lo merecía, brindándome siempre ese hombro de amiga, por criticarme, comprenderme y apoyarme. En mi mente siempre estarás presente y sé que nos seguiremos viendo las caras ya que voy a ser la madrina de tu matrimonio.

A mi tutora Mailen Edith por todo el tiempo dedicado.

A Silvio (Caribeño) que sin él no hubiese sido posible alcanzar bibliografías, en cualquier ocasión estuvo allí brindándome su apoyo incondicional, un gran abrazo para ti y se te quiere.

A los profesores Adrián Naranjo, Joisel Pérez y Erich Mario Gómez por su contribución.

A todos los que de una manera u otra compartieron conmigo e hicieron que mi estancia en la universidad fuese divina.

RESUMEN

En la actualidad es creciente la necesidad de las organizaciones de velar por la calidad de sus datos como fuente fundamental para los análisis en la toma de decisiones, por lo que es de vital importancia contar con procedimientos que ayuden en el proceso de limpieza de datos. Surge este trabajo como resultado de una investigación intensiva sobre el tema, el cual se basa en el análisis de la información de la base de datos del GESPRO, y la identificación de los problemas de calidad de datos que presentan. Para ello se estudian los principales conceptos relacionados con el tema, algoritmos y metodologías para realizar la limpieza de datos.

Mediante el uso de la metodología propuesta por Leslie M. Tierstein se realiza el proceso de limpieza de datos. Se implementan las funciones de similitud de Jaro para la detección y corrección de duplicados y la distancia de edición para los errores ortográficos, no estandarización de cadenas e irregularidades. Implementados por medio de funciones SQL, estos métodos podrán ser corridos en la base de datos erradicándose estos errores de manera automática. Finalmente se valida la eficiencia de dichos métodos a través del cálculo de la complejidad y efectividad, definiéndose para ello un caso de estudio donde se miden en diferentes volúmenes de datos, el tiempo de ejecución y porciento de errores erradicados.

PALABRAS CLAVE

Calidad de datos, limpieza de datos, algoritmo

TABLA DE CONTENIDO

INTRODUCCIÓN

La informática más que una herramienta, es una ciencia que se ha desarrollado a lo largo de los años. Este avance ha logrado resolver diversos problemas a los que se enfrenta el hombre en su vida cotidiana, lo que ha captado la atención de muchas empresas e instituciones que buscan una mejora en los procesos de toma de decisiones. El uso y desarrollo de esta ciencia se hace cada vez más necesario para poder analizar las grandes cantidades de información generadas en los procesos empresariales.

La forma más común de representar computacionalmente información son los datos. En el Diccionario de la Real Academia Española se definen los datos como un antecedente necesario para llegar al conocimiento exacto de algo o para deducir las consecuencias legítimas de un hecho[RAE 2012].

En la actualidad se utilizan varias técnicas para representar y almacenar datos; un ejemplo de ello son las bases de datos (BD) las cuales se definen como un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora, o sea, que una BD puede considerarse una colección de datos variables en el tiempo. Las BD permiten de forma muy sencilla y segura el almacenamiento y obtención de los datos.

Actualmente, las BD son un eslabón fundamental en la planeación y la toma de decisiones en todo tipo de instituciones, no obstante, comúnmente se detecta que los datos almacenados presentan errores que pueden conducir a decisiones erróneas. Según [Dasu et al. 2003] se pone de manifiesto que “es bastante común que las bases de datos tengan del 60% al 90% de problemas de calidad en los datos” lo cual conlleva a la toma de decisiones inadecuadas, y por tanto trae consigo graves problemas como pérdida de tiempo, dinero y credibilidad.

La existencia de “datos sucios” como también se le frecuenta llamar a los errores en las bases de datos suele acarrear grandes dificultades en las instituciones que no se preocupan por la confiabilidad de sus datos, lo que se refleja en el alto costo operacional, la toma de decisiones inadecuada, el incremento de la inseguridad y una desviación de la atención de las direcciones de las instituciones[Hernández 2008].

Existen una variedad de soluciones y aplicaciones, sobre todo en el área de inteligencia artificial (IA), que ayudan o realizan el análisis y procesamiento de los datos en aras de identificar conocimiento que sirva de ayuda a la toma de decisiones.

La IA tiene tanta antigüedad como la informática y ha generado ideas, técnicas y aplicaciones que han permitido resolver problemas difíciles. La IA abre muchas posibilidades para desarrollar sistemas de tratamiento de datos más precisos y robustos[UC3M 2011].

La UCI como una de las casas de altos estudios que reúne mayor potencial de capital humano en la rama de la informática no se ha quedado atrás en el proceso de informatización, tanto de las empresas cubanas como de su propio entorno. En la misma se hace uso de una suite de herramientas llamada GESPRO para la gestión de los proyectos de la universidad. Esta herramienta recoge un conjunto de datos que caracterizan el quehacer de los estudiantes y profesores, además de ser usada en el ámbito de la producción de software; realizándose en ella la asignación de tareas a ejecutar por los mismos. El análisis de este volumen de datos para la identificación de conocimiento que ayude a la toma de decisiones en la gestión de proyectos es una de las ramas de investigación que desarrolla hoy la universidad. Como parte de estas investigaciones surge la creación de una base de conocimientos para la evaluación de competencias genéricas a partir de evidencias.

Esta base de conocimientos forma parte de una pirámide de investigación cuya finalidad es la creación de un sistema experto que evalúe equipos, pero en sí solo recoge las evidencias presentes en cada uno de los sistemas de información utilizados en la universidad, que aportan conocimiento sobre el nivel de las competencias genéricas de ingenieros informáticos para el entorno de la universidad.

Además de la evaluación de las competencias otros de los objetivos que se persigue con la creación de esta base de casos, es precisamente guardar un conjunto de casos sobre los cuáles se puedan realizar procesos de inferencia de conocimiento para descubrir reglas o patrones que ayuden en la toma de decisiones. Sin embargo, para poder realizar inferencias correctas sobre este conocimiento guardado se debe de contar con datos correctos, y el nivel de errores encontrado hoy en las BD primarias de donde son recolectadas las evidencias no permite contar con datos de una calidad adecuada. Entre las fuentes primarias con más errores se encuentra la base de datos del GESPRO, la cual además brinda la mayor cantidad de evidencias a la base de casos.

En dicha BD existen errores como: datos nulos, valores ausentes y la no estandarización de cadenas, refiriéndose este último a la existencia de dos tuplas que se refieren al mismo elemento pero están expresadas en formas diferentes.

Actualmente se necesita contar con la confiabilidad en los datos de la BD del GESPRO; ellos arrojan resultados o decisiones fundamentales para la futura creación de la base de conocimientos. Por tanto para que las inferencias del conocimiento sobre la base de casos sean correctas es necesario que haya calidad en la información guardada en esta BD, para lo que se necesita la limpieza de datos. El proceso de limpieza o corrección de datos va a mejorar el proceso de toma de decisiones, al contar con la información correcta para las mismas, además proveerá la satisfacción del personal.

Por lo anteriormente planteado surge como **problema a resolver:** ¿Cómo mejorar la calidad de los datos recopilados por el GESPRO para la creación de una base de conocimientos?

El **objeto de estudio** se centra en el proceso de limpieza de datos, mientras que el **campo de acción** está enmarcado en el proceso de limpieza de datos de la BD del GESPRO.

Como **objetivo general** del trabajo de diploma se define la implementación de una propuesta de algoritmos que permitan realizar la limpieza de los datos recopilados por el GESPRO.

El objetivo general se desglosa en los siguientes **objetivos específicos**:

1. Elaborar el marco teórico a partir de un análisis crítico del proceso de limpieza de datos.
2. Identificar los algoritmos para realizar la limpieza de datos.
3. Implementar los algoritmos de distancia de edición y similitud de Jaro para realizar la limpieza de datos.
4. Validar la eficiencia de los algoritmos implementados.

Para cumplir el objetivo del trabajo serán necesarias varias **tareas de investigación**:

1. Caracterización de la situación problémica.
2. Redacción del diseño teórico y metodológico de la investigación.
3. Identificación de los principales conceptos a tratar en la investigación.
4. Caracterización de los errores más comunes en los datos y las taxonomías utilizadas.
5. Análisis y caracterización de los algoritmos para resolver problemas de calidad en los datos.

6. Caracterización de las herramientas para la limpieza de datos.
7. Caracterización de las herramientas, metodologías y lenguajes a utilizar en la solución.
8. Análisis de los datos a los cuáles se les va a realizar la limpieza.
9. Implementación de algoritmos de distancia de edición y similitud de Jaro para erradicar los errores identificados.
10. Validación de la implementación realizada.
11. Validación de la mejora en la calidad de los datos.

Se tiene como **idea a defender** que si se implementa un algoritmo de limpieza de datos se logrará mejorar la calidad de los datos en la BD del GESPRO para la creación de una base de conocimiento.

Para el desarrollo completo del trabajo y su total entendimiento se hizo necesario emplear métodos de investigación tales como:

- Histórico lógico: Se utilizará para el estudio de la evolución y desarrollo del proceso de limpieza de datos. Se llevará a cabo una investigación haciéndose un análisis de la evolución del proceso de limpieza de datos mediante la revisión de los artículos y libros que se han publicado del tema, desde lo clásico hasta las nuevas tendencias.
- Analítico-sintético: Se utilizará para el procesamiento de información y elaboración de conclusiones. Este método servirá para analizar y comprender la teoría y documentación relacionada con el tema de investigación, permitiendo así extraer los elementos más importantes relacionados con el objeto de estudio.

El presente documento se estructura en tres capítulos, además de las conclusiones, recomendaciones, y referencias bibliográficas.

Capítulo 1: Fundamentación Teórica.
En este primer capítulo son definidos los conceptos y características que facilitan la comprensión del proceso de limpieza de datos. Además se describen las herramientas y métodos más utilizados en este medio, argumentándose la selección de cada una de ellas para la solución.

Capítulo 2: Proceso de limpieza de datos.

En este segundo capítulo se implementarán los algoritmos de limpieza de datos distancia de edición y similitud de Jaro, que permitan mejorar la calidad de los datos de las bases primarias del GESPRO así como la aplicación de los mismos según la metodología seleccionada.

Capítulo 3: Validación de la propuesta de solución.

En este tercer capítulo se realizará el experimento para validar la eficiencia de los algoritmos implementados. Para ello se efectuarán pruebas a los algoritmos en distintos volúmenes de datos, midiéndose el tiempo de ejecución y porciento de errores erradicados, además de indicarse la efectividad de ellos en el proceso de limpieza de datos.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

En este capítulo son definidos los principales conceptos de la investigación. Se caracteriza el proceso de limpieza de datos, describiéndose las taxonomías utilizadas en Cuba y el mundo, así como los métodos más usados para erradicar los distintos tipos de errores así como las metodologías y herramientas utilizadas para realizar el proceso. Como parte de la solución se describen las funciones de similitud de caracteres y las herramientas a utilizar para implementar los algoritmos.

1.1 Base de casos para la evaluación de competencias a partir de evidencias

El vertiginoso desarrollo de las tecnologías de la información y su incorporación como un elemento fundamental en casi todas las esferas del marco social, hacen que actualmente el uso de las TIC constituya un elemento de vital importancia para la superación y desarrollo de un país. Por tanto, las empresas y organizaciones actuales se ven obligadas a prepararse en áreas de la informática y las telecomunicaciones para poder enfrentar con éxito los retos que impone el nuevo entorno económico mundial.

Al mismo tiempo los sistemas empresariales han ido evolucionando, pasando de los análisis estadísticos de los datos a identificar patrones, construir modelos y extraer conocimiento de ellos. Por tanto, hay que darle a los datos la importancia que se merecen, al menos, tras la aparición de las tecnologías que se pueden englobar en la extensa área del Tratamiento Inteligente de la Información, las cuáles tratan de darle sentido.

Cuba no se ha mantenido alejada de esta realidad. La informatización llevada a cabo en el país hacen que muchas de las empresas tengan soluciones informáticas para gestionar sus procesos o las mismas estén en vías de desarrollo, siendo la UCI uno de los pilares fundamentales para la informatización de la sociedad. Esto ha devenido en que en las organizaciones se estén recopilando gran cantidad de información, que mediante técnicas de inteligencia artificial es utilizada por investigadores y directivos para mejorar la toma de decisiones.

En este sentido, en la UCI con el uso del GESPRO como suite de herramientas para la gestión de los proyectos, y unido al gran número de proyectos y usuarios que trabajan en los mismos, se genera un gran

volumen de datos que caracterizan el quehacer productivo de la universidad. Este volumen de datos es utilizado actualmente para investigaciones que asocian técnicas de la IA con áreas del conocimiento de la gestión de proyectos para obtener nuevos conocimientos, productos y servicios que ayuden en la gestión de los proyectos de la universidad. Tal es el caso de la creación de una base de casos para la evaluación de competencias a partir de evidencias.

Esta base de conocimientos tiene como objetivo guardar un conjunto de casos sobre los cuales se puedan hacer procesos de inferencia que ayuden a evaluar las competencias de las personas dentro de un equipo de desarrollo de software.

En el contexto de la investigación realizada para crear la base de conocimientos [Escobar et al. 2012] se definió como competencia el concepto planteado por la Norma Cubana del Sistema de Gestión Integrada de Capital Humano en el año 2007 la cual reafirma en la consideración de competencias "conjunto sinérgico de conocimientos, habilidades, experiencias, sentimientos, actitudes, motivaciones, características personales y valores, basado en la idoneidad demostrada, asociada a un desempeño superior del trabajador y de la organización en correspondencia con las exigencias técnicas, productivas y de servicios. Es requerimiento esencial que estas competencias sean observables, medibles y que contribuyan al logro de los objetivos de la organización"[Normalización 2007]. Siendo competencias genéricas para ingenieros informáticos, aquellas que "abarcan los comportamientos asociados con desempeños comunes a diversas ocupaciones y ramas de actividad productiva, además de ser exigibles en mayor o menor grado a todo profesional y estudiante de la universidad" [Pérez Quintero 2010] y evidencias aquellos "datos o información que están presentes en los sistemas que se usan en la universidad y que caracterizan el comportamiento de una persona y por tanto brindan conocimiento que puede permitir identificar el nivel de una determinada competencia poseída por esa persona"[Escobar, et al. 2012].

Para lograr evaluar las competencias, se recogen dentro de cada caso evidencias aportadas por los sistemas de información utilizados en la universidad. La Tabla 1 muestra la cantidad de evidencias aportadas por cada sistema de gestión a cada una de las 12 competencias a evaluar.

Tabla 1. Total de evidencias aportadas por cada herramienta de gestión [Escobar, et al. 2012].

Sistemas origen	**Aporte a las competencias**	**Puntuación**

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	Total
Sistema de gestión de proyectos.	2	3	9	2	1	2	8	3	3	3	7	6	49
Sistema de gestión de versiones.	2	1	6	4	0	0	0	0	0	0	0	2	15
Sistema Encuestas disponibles.	4	3	1	1	3	0	2	5	1	3	4	4	31
Total	8	7	16	7	4	2	10	8	4	6	11	12	95

Como se puede observar en la tabla anterior, el sistema de gestión de proyectos GESPRO es el que más evidencia aporta, de ahí la importancia de realizar un proceso de limpieza que eleve la calidad de los datos a utilizar en la base de casos.

1.2 Principales conceptos.

- **Calidad de los datos.**

Para cualquier análisis crítico o aplicación la principal tarea es la mejora de la calidad de los datos, sin embargo como las fuentes son diversas, los formatos heterogéneos y el volumen de los datos crece rápidamente, el mantenimiento y la mejora de los datos se hace más difícil[Redman 2001].

En la literatura, la calidad de los datos es presentada como un concepto multidimensional [Wang and Strong 1996]. Una de las definiciones más completas está expresada en una colección de dimensiones organizadas dentro de 4 categorías (Tabla 2).

Tabla 2: Categorías y dimensiones de la calidad de datos[Wang and Strong 1996].

Categorías de la Calidad de los Datos	Dimensiones de la Calidad de los Datos
Intrínseco	Precisión, Objetividad, Credibilidad, Reputación
Accesibilidad	Accesibilidad, Seguridad de acceso
Contextual	Relevancia, Valor añadido, Oportuno, Integridad, Cantidad de datos
Representativo	Interpretable, Fácil de entender, Representación concisa, Representación consistente

Calidad de datos intrínseca: no solo incluye a la precisión y objetividad (esto es evidente para profesionales de sistemas de información), sino también credibilidad y reputación. Esto sugiere que,

contrario a la vista de desarrollo tradicional, los consumidores de datos también ven la reputación y la credibilidad como parte integral de la calidad de datos intrínseca, debido a que la precisión y la objetividad por sí solas no son suficientes para que los datos tengan una buena calidad.

Calidad de datos accesible: los usuarios finales ven esta categoría como un importante aspecto en la calidad de los datos. No obstante, existe una diferencia entre tratar a la accesibilidad como una categoría de la calidad de datos global, o separándola de otras categorías de la calidad de datos, en cualquiera de los casos debe ser tomada en cuenta.

Calidad de datos contextual: no está explícitamente reconocida en la literatura sobre la calidad de los datos. Esta agrupación de las dimensiones de la calidad de datos contextual revela que la calidad de los datos debe ser considerada dentro del contexto de la tarea a mano.

Calidad de datos representativa: incluye aspectos relacionados con el formato de los datos (representación concisa y consistente) y su significado (interpretables y fáciles de entender).

En el presente trabajo la calidad de los datos se vio identificada en la dimensión representativa, ya que una vez realizado el proceso de limpieza, estos deben mantener un formato interpretable y fácil de entender, contando con una representación concisa y consistente; o sea la aplicación de dicho proceso no debe afectar de ninguna manera la integridad de la información almacenada.

Según [Müller and Freytag 2003] los datos tienen que satisfacer un conjunto de criterios de calidad. A partir de los criterios de calidad se puede definir la forma de evaluar las puntuaciones para una colección de datos existente. (Figura 1)

Figura 1: Jerarquía de criterios de calidad. [Müller and Freytag 2003]

Para medir la calidad de una colección de datos, los resultados tienen que ser evaluados para cada uno de los criterios de calidad. La evaluación de las calificaciones de los criterios de calidad se puede usar para cuantificar la necesidad de limpieza de datos para una colección de datos, así como el éxito de un proceso de limpieza de datos realizado en una colección de datos. Los criterios de calidad también se pueden utilizar dentro de la optimización de la limpieza de datos por las prioridades que especifican para cada uno de los criterios que a su vez influye en la ejecución de datos.

A partir de lo antes planteado las autoras coinciden con [Müller and Freytag 2003] en que la calidad datos se define como un valor agregado sobre un conjunto de criterios de calidad y que a partir de estos se describe que el conjunto de criterios se ve afectado por la limpieza integral de datos y la definición de la forma de evaluar las puntuaciones para cada uno de los datos existente.

No obstante en el desarrollo del presente trabajo, luego de estudiados varios criterios y categorías de calidad, las autoras definen evaluar la calidad de los datos a través de la efectividad de los algoritmos implementados, pues con estos se logrará disminuir la cantidad de errores presentes en los mismos y de esta forma elevar la calidad de los datos.

- **Limpieza de datos.**

La limpieza de datos según[López 2011]es un proceso que se caracteriza por detectar y corregir los errores en los datos.

Por otra parte [Müller and Freytag 2003]definen la limpieza de datos como la totalidad de las operaciones realizadas a los datos para eliminar anomalías y recibir una colección de datos siendo una representación exacta y única.

En[Hernández 2008]se expresa que limpieza de datos no es más que corregir o remover información incorrecta, con formato inapropiado o duplicado en una base de datos a través de métodos computarizados. En general, el campo de trabajo de la limpieza de datos se centra en:

- Definir y determinar los tipos de errores.
- Buscar y determinar instancias con error.
- Corregir los errores.
- Documentar las instancias con errores y los tipos de errores.
- Modificar los procedimientos de entrada de datos para reducir errores futuros.

En conclusión las autoras definen la limpieza de datos como el acto de detección, corrección o eliminación de datos erróneos de una base de datos. El proceso de limpieza de datos permite identificar datos incompletos, incorrectos, inexactos, no pertinentes, y luego sustituir, modificar o eliminar estos datos.

1.3 Tipos de anomalías de datos y taxonomías.

La taxonomía es una manera para clasificar, nombrar y agrupar en categorías, dependiendo de las características, para analizar de manera distinta y normalizar la denominación de los datos.

La anomalía es una propiedad de los valores de los datos, a partir de la cual se obtiene una representación errónea del mini mundo que éstos reflejan. Esta puede ser originada por diferentes causas.

A los datos que contienen anomalías se les denomina datos erróneos o sucios y su presencia puede obstaculizar el uso efectivo y eficiente de la información[Rahm and Do 2000].

En conclusión una taxonomía es un conjunto en el que se clasifican las anomalías, estas por separadas procesan los datos de maneras distintas de acuerdo a sus similitudes, pero al final manipulan la información de acuerdo al punto de vista planteado por dicha taxonomía.

Según la taxonomía propuesta por [Kim et al. 2003]existen tres formas de clasificar los errores:

- **Datos que faltan**: faltante de información, datos nulos y vacíos.
- **Datos que no faltan:** datos erróneos debido a:
    - No observancia de las restricciones de integridad de forma automática exigibles (limitaciones de usuarios específicos (datos que cuelgan, datos duplicados y datos mutuamente contradictorios) y las restricciones de integridad no soportados en los sistemas de bases de datos (nivel de abstracción equivocado, datos temporales no actualizados)).
    - No ejecutabilidad de restricciones de integridad (error de datos en el ingreso que implica una sola tabla o archivo (error de datos en uno o múltiples campos) y la inconsistencia de valores en varias tablas de un mismo atributo)
- **Datos correctos, pero inutilizables**:
    - Datos diferentes para la misma entidad (la diferencia de valores para una misma entidad en tablas diferentes o bases de datos diferentes).
    - Datos ambiguos (uso de abreviaturas y contexto incompleto).
    - La disconformidad de la estandarización de datos (la representación diferente de los datos compuestos (datos concatenados y jerárquicos) y los no compuestos (la transformación de algoritmos (formatos de codificación de ASCII, EBCDIC, representaciones incluyendo números negativos, moneda, fecha, fracción, etc. y las unidades de medición incluyendo hora, distancia, peso, área y volumen) y la no transformación de algoritmos (abreviaturas, alias))).

Según Beatriz[López 2011]las anomalías se clasifican en:

- **Datos Incompletos**: Cuando no se almacena la información o existen campos que no se utilizan en las bases de datos.

- **Datos Incorrectos:** Cuando utilizan códigos incorrectos, existen informaciones repetidas que se refiere a un mismo elemento, fallos tipográficos o no continuar el patrón establecido.

- **Incomprensibles:** Cuando existen valores de atributos diferentes.

- **Inconsistentes:** Cuando existe el cambio de codificadores y en las bases de datos todavía conservan los codificadores antiguos, diversos códigos con el mismo significado, uso inconsistente de nulos, vacíos y espacios para indicar la ausencia de información y la no correspondencia entre la llave extranjera y la primaria.

Se tomó para la investigación la taxonomía propuesta por[Hernández 2008], por ser la que trata de forma más específica las anomalías presentadas en la BD en cuestión, esta divide a las anomalías en:

- **Anomalías sintácticas:** son aquellas relativas a las alteraciones que se pueden producir en las gramáticas, que impiden la correcta formación de oraciones y conceptos, o que afectan a las reglas que definen las secuencias correctas de los elementos de un lenguaje. Este tipo de anomalías se pueden dar en tres formas.

    - **Errores léxicos**: Son aquellos errores que se dan en el vocabulario, conjunto de palabras que pertenecen a un campo semántico dado en los que la estructura de los datos difiere del formato especificado para dichos datos, es decir, el número de valores es inesperado (mayores o menores) para una tupla t o, el grado de una tupla #t es diferente de #R, el grado del esquema de relación previsto para la tupla.

    - **Errores en el formato del dominio**: Los errores en el formato o las características de un valor son aquellos donde el valor dado por un atributo A no se ajusta con el dominio u orden determinado del formato previsto anteriormente G (dom(A)).

- **Irregularidades**: Son los errores relacionados con el uso no uniforme de valores, unidades (de medida, de peso, etc.) y abreviaturas. Esto pasa por ejemplo, si usamos diferentes tipos de monedas para especificar el salario de los empleados de una determinada empresa, lo cual se convierte en un problema mayor si los tipos de moneda no son explícitamente listados con cada valor correspondiente. Esto resulta en valores con una representación correcta de los hechos si tenemos el conocimiento necesario acerca de su representación para poder interpretarlos. Otro ejemplo es el uso o el diferente uso de las abreviaturas.

- **Anomalías semánticas:** son aquellas discrepancias de una regla relativas a la significación de las palabras, de ellas se dan cuatro casos:

  - **Violaciones de las restricciones de integridad:** Son aquellas en las que se describen tuplas (o grupos de tuplas) que no satisfacen una o más de las restricciones de integridad. Las restricciones de integridad son usadas para describir el entendimiento del mini-mundo mediante el conjunto de instancias válidas. Cada restricción es una regla que representa el conocimiento acerca del dominio y los valores permitidos para una representación certera de los hechos.

  - **Contradicciones:** Las contradicciones son violaciones de la dependencia funcional que pueden ser representadas como restricciones de integridad o duplicados con valores inexactos, son valores dentro de una tupla o entre tuplas que violan algún tipo de dependencia entre valores. Un ejemplo del primer caso pudiera ser una contradicción entre el atributo EDAD y FECHA_NACIMIENTO para una tupla que representa personas.

  - **Duplicados:** Es cuando dos o más tuplas representan la misma entidad del mini-mundo. Los valores de estas tuplas no necesariamente deben ser idénticos. Los duplicados inexactos son casos específicos de contradicción entre dos o más tuplas, ellos representan la misma entidad pero con diferentes valores para todas o algunas de sus propiedades. Esto dificulta la detección de duplicados y su fusión.

  - **Tuplas no válidas:** Éstas representan la anomalía más complicada encontrada en colecciones de datos. Por tuplas no válidas queremos decir aquellas tuplas que no muestran anomalías del tipo de las definidas anteriormente pero todavía no representan entidades válidas del mini-

mundo. Las tuplas no válidas pueden además representar excepciones y por consiguiente no deben considerarse como errores.

- **Anomalías de alcance.** Se evidencian de dos formas:
    - **Valores que faltan**: Son el resultado de omisiones en el proceso de colección de los datos.
    - **Tuplas que faltan**: Es el resultado de las omisiones de entidades completas existentes en el mini mundo que no son representadas por tuplas en la colección de datos [Redman 2001].

A pesar de que en el GESPRO ya se le dan solución mediante funciones implementadas por el equipo del departamento a las anomalías de alcance, se presentan tipos de errores como irregularidades y duplicados**;** siendo el objetivo del presente trabajo implementar una propuesta de algoritmos para la solución a dichos problemas.

1.4 Algoritmos de limpieza de datos.

En la actualidad las BD se ven expuestas a la ocurrencia de errores, además de la pérdida e inconsistencia en los datos, esto se debe a los grandes volúmenes de datos que se manejan hoy en día. Para esto existen numerosos algoritmos, métodos y técnicas, que pueden ser aplicadas para eliminar los errores y corregir las inconsistencias.

Se exponen los siguientes algoritmos o métodos para erradicar errores en los datos:

- **Transformación de datos. (Data transformation)**

La transformación de los datos propone transformar los datos del formato dado a un formato esperado por la aplicación, esto involucra el esquema de tuplas y el dominio de sus valores. El esquema de transformación es desarrollado en el proceso de limpieza de datos, los cuales son transformados en un esquema común de acuerdo a las necesidades de la aplicación. La corrección de los valores debe ser desarrollada solo en casos donde los datos de entrada no se corresponden con el esquema y puedan llevar a fallos posteriores en el proceso de transformación[Marmol and López 2005]. Esto hace que la

limpieza de datos y el esquema de transformación se vean como tareas suplementarias. Puede involucrar las siguientes técnicas[Han and Kamber 2000]:

- Mitigador (Smoothing): Facilitar el trabajo de eliminación de los datos con errores, que puede incluir agrupamiento y regresión.

- Agregación (Agregation): Son aplicadas operaciones de resumen o agregación de los datos. Este paso es usado generalmente en la construcción de cuadros de datos para el análisis de los datos en múltiples granularidades.

- Generalización (Generalization): Donde los datos de bajo nivel o primitivos son sustituidos por conceptos de alto nivel a través del uso de conceptos jerárquicos. Por ejemplo tenemos los atributos categóricos tales como calle que puede ser generalizado en un concepto de más alto nivel como ciudad o provincia, también en valores de atributos numéricos como edad que puede ser generalizado en conceptos de alto nivel como: joven, adulto o anciano[Han and Kamber 2000].

- Normalización (Normalization): La normalización es una transformación en el nivel de instanciación utilizada con la intención de eliminar irregularidades en los datos. Esto incluye conversión de valores simples o funciones de traducción, así como, normalizar valores numéricos que están en un intervalo fijo dado por un valor máximo y un mínimo [Abiteboul et al. 1999; Marmol and López 2005; Sattler et al. 2000],donde los atributos de los datos están en una escala dentro de un rango determinado, por ejemplo, -1.0 a 1.0, 0 a 1.0.

- Construcción de atributos (Attribute construction or feature construction): Cuando los atributos son construidos y adicionados desde un conjunto de atributos dado, para ayudar en el proceso de minería de datos.

- **Eliminación de duplicados. (Duplicate Elimination)**

El algoritmo de eliminación de duplicados ordena todos los atributos en grupos de tuplas similares; donde cada tupla debe ser comparada con el resto mediante métodos de detección de duplicados, que se mencionan a continuación:

- Método de vecindad ordenada (Sorted Neighbourhood Method)[Hernández 1995]: es un método rápido, que reduce el número de comparaciones requeridas mediante el ordenamiento de las tuplas por una llave construida de los atributos de la relación, que brinda los duplicados cerca unos de otros, entonces solo se comparan las tuplas que están en un mismo grupo. La identificación de tuplas duplicadas se hace usando reglas basadas en el conocimiento específico del dominio. Para mejorar la precisión, los resultados de varios pases de la detección de duplicados pueden ser combinados con el cálculo de cercanía transitiva de todos los pares de tuplas duplicadas encontradas.

- Eliminación de duplicados borrosos (Fuzzy Duplicates): este método propone una solución que evita los problemas de los métodos de ordenamiento, los cuales confían en las dimensiones jerárquicas típicamente asociadas con las tablas dimensionales en un almacén de datos. Éstas son jerarquías de tablas típicamente en relaciones 1-n expresadas por relaciones entre llaves (de llaves extranjeras). Cada tupla en la relación 1 es asociada con un grupo de tuplas de la relación n. El grado de solapamiento entre grupos asociados con dos tuplas de la relación 1 es una medida de la co-ocurrencia entre ellos, y puede ser usada para detectar duplicados.

- **Análisis gramatical. (Parsing)**

El análisis gramatical es desarrollado para la detección de errores sintácticos mediante analizadores gramaticales, los cuales deciden si una cadena dada es un elemento del lenguaje definido. En el proceso de limpieza de datos las cadenas son tuplas completas de una instancia relacional o valores de atributos de un dominio. La existencia de un gran número de errores sintácticos en una colección de datos depende de la extensión del esquema aplicado en el ambiente donde los datos son mantenidos. Si los datos son salvados en ficheros planos existe la posibilidad de errores léxicos y de dominio, en este caso se usa una gramática derivada de la estructura del fichero. Los datos son manejados por sistemas de administración de bases de datos, que no esperan que los datos contengan errores léxicos o de dominio, pero los errores de este tipo pueden existir para cada uno de los atributos [AHO and ULLMAN 1979; Raman and Hellerstein 2001].

- **Aplicación de las restricciones de integridad. (Integrity Constraint Enforcement)**

La aplicación de las restricciones de integridad describe el problema de garantizar el cumplimiento de las restricciones después de transacciones, ya sea modificando la colección de datos, insertando, borrando o actualizando tuplas. Las dos soluciones son: chequeo de las restricciones de integridad y mantenimiento de las restricciones de integridad. El chequeo de las restricciones de integridad rechaza las transacciones que si se efectúan pueden violar alguna restricción de integridad, mientras que el mantenimiento de las restricciones de integridad tiene que ver con la identificación de las actualizaciones adicionales (por ejemplo: reparaciones), para ser añadidas a la transacción original y garantizar que la colección de datos resultante no viole alguna restricción de integridad[Mayol and Teniente 1999].

- **Métodos estadísticos. (Statistical Methods)**

Los métodos estadísticos pueden ser usados para la verificación de los datos, así como en la detección y corrección de anomalías. La detección y eliminación de errores que representan tuplas no válidas va más allá del chequeo y la aplicación de las restricciones de integridad, a menudo involucran relaciones entre dos o más atributos que son difíciles de descubrir y describir por las restricciones de integridad. Esto puede ser visto como un problema en la detección perfilada, como por ejemplo, una minoría de las tuplas y valores que no están acordes a las características generales de una colección de datos dada[Hernández 2008].

Una posible solución incluye métodos estadísticos que calculan algún valor medio o medida estadística. Existen valores que pueden ser detectados como violaciones de las reglas de asociación u otros patrones existentes en los datos[Hernández 2008]. Otra anomalía manipulada por los métodos estadísticos son los valores que faltan, estos valores son manipulados basados en la entrada de uno o más valores posibles[Maletic and Marcus 2005].

Para la implementación de la solución no se seleccionaron algoritmos de este tipo. Estos tienen un alto grado de eficiencia, pero están basados en técnicas avanzadas de inteligencia artificial, o en errores que se cometen de forma más o menos habitual. Se decidió en este trabajo de diploma realizar implementación de algoritmos basados en funciones de similitud basadas en caracteres, los cuales son más manejables y hay más disponibilidad de información en los medios.

1.4.1 Herramientas para la limpieza de datos.

Para el desarrollo de la solución se analizaron herramientas de limpieza de datos, haciendo énfasis en los algoritmos que son implementados por ellas y los tipos de errores que estas corrigen, a estas se hace referencia por ser las más utilizadas en el mundo para la tarea de limpieza de datos.

Dentro de las herramientas más utilizadas para erradicar los tipos de errores en las BD se estudiaron las siguientes:

- **Arktos**: es un armazón (framework) capaz de modelar y ejecutar el proceso de Extracción-Transformación-Carga (ETL Process (Extraction-Transformation-Load)) para la creación de un almacén de datos (data warehouse). La limpieza de datos es una parte integral de este proceso de ETL el cual consiste en simples pasos para extraer datos relevantes de la fuente, transformarlos en el formato destino y limpiarlos, para luego cargarlos dentro del almacén de datos. [Vassiliadis et al. 2001].

- **IntelliClean:** es un software de limpieza de datos basado en reglas con un mayor enfoque en la eliminación de duplicados. El framework propuesto consta de tres etapas. En la etapa de pre-procesamiento los errores sintácticos son eliminados y los valores estandarizados en formato y consistencia del uso de abreviaturas. La etapa de procesamiento representa la evaluación de las reglas de limpieza en los datos condicionados que especifican acciones a tomar bajo determinadas circunstancias. Existen cuatro tipos de reglas:

    - Las reglas de identificación de duplicados especifican las condiciones bajo las cuales las tuplas se consideran como duplicadas.

    - Las reglas de fusión/depuración especifican como las tuplas duplicadas van a ser manipuladas.

    - Las reglas de actualización especifican la forma en que los datos van a ser actualizados en una situación particular, esto habilita la especificación de las reglas de puesta en vigor de las restricciones de integridad, para cada restricción de integridad se define una regla de actualización que define como modificar la tupla para así satisfacer la restricción [Lee et al. 2000].

- **AJAX:** es un framework flexible y extensible que intenta separar los niveles lógicos y físicos de la limpieza de datos. El nivel lógico sustenta el diseño del flujo de trabajo de la limpieza de datos y la especificación de las operaciones de limpieza desarrolladas, mientras que en el nivel físico recae su implementación. El mayor interés de AJAX es transformar los datos existentes de una o más colecciones de datos en un esquema destino y eliminar los duplicados dentro de este proceso. Para este propósito, se define un lenguaje declarativo basado en un grupo de operaciones de transformación, las cuales son: asignación (mapping), visión (view), correspondencia (matching), agrupación (clustering) y fusión (merging) [Galhardas et al. 2000; Galhardas et al. 2001].

- **Potter´s Wheel:** es un sistema interactivo de limpieza de datos que integra la transformación de datos y la detección de errores usando una hoja de cálculo como interfaz. Los efectos de las operaciones desarrolladas son mostrados inmediatamente en tuplas visibles en pantalla. La detección de errores es hecha para la colección de datos completa de forma automática como un proceso de fondo. Un grupo de operaciones son especificadas y soportan los esquemas de transformaciones comunes sin una programación explícita. Las especificaciones para el proceso de limpieza de datos son hechas de forma interactiva, además de que la retroalimentación inmediata de las transformaciones llevadas a cabo y las detecciones de errores permiten a los usuarios un desarrollo gradual y pulir el proceso[Raman and Hellerstein 2001].

- **FuzzyDupes 2007:** Es una herramienta con licencia Shareware y su autor es Kroll Software-Development; esta herramienta tiene como objetivo la búsqueda de duplicados borrosos. Para este propósito se utilizan algoritmos que comparan cadenas de caracteres y detectan patrones recurrentes dentro de esas cadenas, más conocidos como algoritmos de similitudes de patrones (pattern matching algorithms), un ejemplo de ellos es la métrica de similitud (también conocido como Levenshtein distance metric). Este algoritmo muestra el número fundamental de pasos (insertar, modificar, eliminar) necesarios para convertir una cadena de caracteres A en una cadena de caracteres B. Como el número de comparaciones a hacer se incrementa gradualmente se intenta agrupar todos los registros antes de iniciar la búsqueda, para así poder hacer las comparaciones incluso en grandes bases de datos[Hernández 2008].

- **ListCleaner Pro:** Es un software para la limpieza de datos y duplicados y para la corrección de bases de datos, hojas de cálculo, correos electrónicos, etc. Puede manejar listas de contactos electrónicos,

listas de direcciones electrónicas, listas de categorías y precios de productos, detalles de ventas y nombres de estudiantes, entre otras. Para buscar duplicados el programa analiza las cadenas de datos y sus comparaciones lógicas, también elimina signos de puntuación indeseados y errores ortográficos e identifica datos faltantes[Hernández 2008].

- **FraQL:** es otro lenguaje declarativo de apoyo a la especificación del proceso de limpieza de datos. El lenguaje es una extensión del SQL basado en un modelo de datos objeto-relacional. Soporta la especificación de esquemas de transformación así como transformaciones de datos a nivel de instancias, como por ejemplo estandarización y normalización de valores. Esto puede hacerse mediante funciones definidas por el usuario, las cuales deben hacerse para los requerimientos específicos del dominio dentro del proceso individual de limpieza de datos[Sattler, et al. 2000; Sattler and Schallehn 2001].Con sus operadores extendidos union y join en conjunción con las funciones de conciliación definidas por el usuario, FraQL maneja la detección y eliminación de duplicados.

- **WinPureClean & Match 2007:** Es un software de limpieza de datos, que puede importar varias listas a la misma vez, incluye opciones avanzadas para la búsqueda de similitudes, provee cinco módulos de limpieza de datos que ayudan a asegurar que las listan están totalmente llenas, que son precisas y que tienen las direcciones correctas, también utiliza algoritmos de similitudes de patrones y utiliza métodos para eliminar el problema de mezcla/limpieza (merge/purge). Es utilizada para limpiar listas de correo electrónico, bases de datos de comercialización, hojas de cálculo y correos electrónicos[Hernández 2008].

En [López 2011]se hace un estudio de las herramientas de limpieza de datos en función de los procesos que realizan las mismas. Ver tabla 3.

Tabla 3: Herramientas de limpieza de datos

Herramienta	Análisis de datos	Estandarización de cadenas	Estandarización de direcciones postales	Reemplazo de valores ausentes
Oracle	Perfil de datos muy completo. Calcula mínimo, máximo, media, cardinalidad, cantidad de ausentes,	Operador Match/Merge Busca tuplas similares y las mezcla en una. Puede ser por nombre, dirección o	Operador Name Address Necesita de licencia e instalación de software producido por terceros con información de	No

	Detecta: dominio, llaves, dependencias funcionales, llaves extranjeras, patrones de datos como expresiones regulares. No reglas de asociación	cadena. Utiliza la similitud entre cadenas para buscar similares. Distancia de Edición y Jaro-Winkler	nombres y direcciones para limpiar y estandarizar Cuba no aparece en sus listas	
RapidMiner	Detección de outlier	No	No	Reemplaza ausentes por mínimo, máximo, valor determinado, promedio Atributos nominales por la moda
Pentahoo Data Integrator	Obtiene el perfil de datos numéricos solamente, calcula cardinalidad, media, desviación, mínimo, máximo	No	No	Reemplaza por un valor determinado. Se puede indicar por tipo de datos o por campos
Data Cleaner	Perfiles de datos amplios: nulos, vacíos por tipo de campo Numéricos: min, max, suma, media, desviación, patrones. Cadenas: longitud máxima, mínima, cantidad de caracteres promedio. Valores más frecuentes	No	No	No
SQL Server 2008	Perfiles de datos muy completos. Estadísticos de los campos numéricos, por ciento de nulos. Determina dependencias funcionales, llaves primarias	Utiliza una transformación llamada Fuzzy Grouping. Crea grupos de cadenas empleando la similitud entre cadenas con la distancia q-gram	No directamente, usa el mismo operador que para cadenas. No las segmenta	No Los trata luego en las técnicas de minería

Estas herramientas son propietarias por lo que no se puede acceder al código; por tanto no se puede obtener ninguna ayuda respecto a cómo estas implementan los algoritmos de limpieza de datos. El GESPRO necesita una solución rápida que no requiera más de una persona para realizarla, y sobre todo que el proceso sea ejecutado de forma automática.

El cliente (GESPRO) no está interesado en la elaboración o uso de una herramienta para la realización del proceso, por lo que se ejecutará la implementación de una serie de algoritmos de limpieza de datos mediante funciones SQL, las cuales serán de fácil manejo para el personal encargado de esta tarea.

1.4.2 Funciones SQL

Una función en PostgreSQL son sentencias SQL agrupadas y pre-compiladas para ejecutarse en bloque dentro del servidor [Márquez 2012], a diferencia de las consultas SQL donde cada consulta es procesada en tiempo de ejecución por el servidor, las funciones procedurales son compiladas cuando son creados, ya que el servidor asume que serán ejecutados más de una vez, una función ofrece las siguientes ventajas:

- No sobrecarga la comunicación cliente/servidor al evitar enviar una consulta tras otra, en su lugar procesa una consulta tras otra y envía únicamente el resultado.
- Cuando se ejecuta la primera vez se crea un plan preparado de ejecución, las siguientes ejecuciones reutilizan el plan preparado.
- Agrega estructuras de control y capacidad de cálculo al lenguaje SQL.
- Las mismas consultas están disponibles para varias aplicaciones.
- Los datos solo están accesibles mediante las funciones y evita el uso de inyecciones SQL.

1.5 Funciones de similitud basadas en caracteres.

La distancia o medida de similitud entre cadenas se refiere a la cantidad de diferencias que hay entre ellas. Es utilizada para solucionar diversos problemas, sobre todo aquellos que tienen relación con el procesamiento de textos o del lenguaje natural. El cálculo de la similitud entre cadenas adquiere gran importancia, sobre todo cuando se trata de bases de datos, las cuales requieren contar con datos de calidad para posteriores inferencias de conocimiento. Estas funciones de similitud consideran cada cadena como una secuencia ininterrumpida de caracteres [Amón 2010]. Entre estas se encuentran:

- Distancia de edición.

- Distancia de brecha afín.
- Similitud Smith-Waterman.
- Similitud de Jaro.
- Similitud de q-grams.

Distancia de edición: la distancia de edición entre dos cadenas 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:

- Reemplazar un carácter de A por otro de B (o viceversa).
- Eliminar un carácter de A o B.
- Insertar un carácter de A en B (o viceversa).

Distancia de brecha afín: la distancia de edición y otras funciones de similitud tienden a fallar identificando cadenas equivalentes que han sido demasiado truncadas, 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 $h<g$ [Gotoh 1982].

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 and Waterman 1981] 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, deben ser descartados. Por ejemplo, "PhD Jorge Eduardo Rodríguez López" y "Jorge Eduardo Rodríguez López, Universidad Nacional de Colombia" tendrían una similitud cercana a uno, pues el prefijo "PhD" y el sufijo "Universidad Nacional de Colombia" serían descartados sin afectar el puntaje final.

Similitud de Jaro: Jaro desarrolló una función de similitud que define la trasposición de dos caracteres como la única operación de edición permitida [Jaro 1976]. Los caracteres no necesitan ser adyacentes, sino que pueden estar alejados cierta distancia d que depende de la longitud de ambas cadenas.

Similitud de q-grams: un q-gram, también llamado n-gram, es una subcadena de longitud q [Yancey 2006]. El principio tras esta función de similitud es que, cuando dos cadenas son muy similares, tienen muchos q-grams en común.

De las funciones de similitud mencionadas anteriormente fueron seleccionadas para su implementación la de distancia de edición, usada para errores ortográficos y tipográficos. Se hará uso de esta función porque es muy usada por los correctores ortográficos y en la detección de errores en listas de datos, para sugerir palabras almacenadas en un fichero o diccionario del sistema, que se asemejen a la detectada con problemas ortográficos según una medida de similitud entre cadenas. Por otra parte para darle solución a los problemas de duplicados se hará uso de la función de similitud de Jaro, por ser este uno de los métodos más usados dentro de este campo, además de su efectividad por encima de otros de los anteriormente mencionados. En el capítulo dos del documento se abordarán con mayor profundidad las características de ambos métodos.

1.6 Metodologías de limpieza de datos

Las metodologías de limpieza de datos buscan estandarizar la forma de trabajo de los integrantes del proyecto para poder lograr los objetivos propuestos. En el desarrollo de la investigación se han estudiado tres metodologías con diferentes enfoques.

Oracle dispone de su herramienta Warehouse Builder para la construcción de bodegas de datos, que incluye un módulo previo para análisis de calidad y facilita un modelo completo de limpieza de datos en sus procesos de ETL. Es decir, el proceso propuesto esta sesgado a una herramienta. Esta metodología, basada en el llamado Ciclo de la Calidad de los Datos (Data Quality Cycle) puede ser adaptada para desarrollarse sin utilizar el Warehouse Builder, pues sugiere el qué hacer, sin necesidad de tener que hacerlo con sus especificaciones técnicas, usando la herramienta [Rochnik and Dijcks 2006].

Rahm y Do, en su distinción entre problemas de una única fuente y problemas de múltiples fuentes, presenta una aproximación sobre cómo alcanzar la calidad de los datos que se trabajan en un proceso de ETL.[Rahm and Do 2000]

Tierstein presenta una metodología muy especializada, enfocada en la transferencia de datos de uno o varios sistemas hacia nuevas bases o bodegas de datos. Se considera que ésta es una de las más completas, pero quizás la cantidad de pasos y algunas actividades que se realizan pueden ser omitidas, y la metodología podría complementarse con otros pasos que pueden ser cruciales para un exitoso proceso de análisis de la calidad de los datos.[Christen 2006; Tierstein 2005]

Basándose en la propuesta realizada por Tierstein, [Paniagua et al. 2010] realizan algunos cambios a la misma, en base a los requisitos que tienen que cumplir en su proyecto, haciendo énfasis en la fase de planeación, perfilamiento y diagnóstico; claridad de los conceptos y de los pasos a seguir; y facilidad de utilización (exigencias de personal, profundidad de la documentación). Teniendo en cuenta estas premisas proponen una adaptación de la metodología que permite realizar el proceso de limpieza de datos con mayor agilidad.

Luego de estudiadas cada metodología se concluye que para el proceso de limpieza actual se considera que la que más se ajusta es la metodología propuesta por James Paniagua [Paniagua, et al. 2010], ya que es la metodología más detallada y con mejor explicación de las analizadas, además de no estar sesgada a ninguna herramienta y no requerir de exceso de documentación. Como está basada en la metodología de Tierstein, recoge las mejores prácticas propuestas por la misma, la cual es muy similar con la de Oracle (que es una metodología más orientada a la herramienta) y la de Rahm y Do (que tiene un enfoque más teórico), por tanto en esta metodología se evidencian elementos de todas las analizadas.

Las fases que propone la metodología son:

1. Planeación y Preparación

    - Contextualización y compresión del negocio

2. Análisis y diseño conceptual

    - Determinación de una muestra de los datos para identificar principales problemas

- Identificación de los problemas en la muestra con base en una taxonomía de problemas comunes.
- Realización de listado de tablas a limpiar

3. Realizar la limpieza
   - Definición de métodos de limpieza para los problemas seleccionados
   - Limpieza de los datos
   - Realización de un informe comparativo del estado de los datos antes y después de la limpieza.
4. Generación de resultados y análisis

1.7 Herramientas

Para el desarrollo de la propuesta de solución se caracterizaron las herramientas necesarias para que la implementación de la misma se realizara de manera ágil y en un entorno adecuado. Dichas herramientas permitirán al programador realizar las funciones y consultas a la BD para el proceso de limpieza de datos. Además, al realizarse la solución para ser implementada en las BD del GESPRO, las herramientas a utilizar deben ser las mismas que ya están definidas por el equipo de desarrollo, en aras de lograr compatibilidad entre la solución realizada y el GESPRO. En función de que para la solución se van a estar implementando los algoritmos mediante funciones SQL las herramientas a utilizar se describen a continuación.

1.7.1 PostgreSQL 9.1

PostgreSQL 9.1 proporciona varias características, herramientas e innovaciones como permitir la alta disponibilidad con consistencia a través de varios servidores, admite la correcta ordenación lingüísticamente por base de datos, tabla o columna, ejecutar en complejas etapas, múltiples actualizaciones de datos en una sola consulta, conectar y consultar otras bases de datos de PostgreSQL, así como crear fácilmente, cargar y administrar las características de base de datos nuevas. Muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos sistemas gestores de bases de datos comerciales [PostgreSQL 2013].

Soporte de tipos y funciones de usuario PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario. Diseñado para ambientes de alto volumen PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC (Multi-Version Concurrency Control /Control de concurrencias para múltiples versiones) para conseguir una mejor respuesta en ambientes de grandes volúmenes. PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. El código fuente está disponible para todos sin costo [PostgreSQL 2013].

1.7.2 PL/pgSQL

PL/pgSQL (Procedural Language/PostgreSQL Structured Query Language) es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las sentencias SQL básicas. Como un verdadero lenguaje de programación, dispone de estructuras de control repetitivas y condicionales, además de la posibilidad de creación de funciones que pueden ser llamadas en sentencias SQL normales o ejecutadas en eventos de tipo disparador (trigger). Las funciones escritas en PL/pgSQL aceptan argumentos y pueden devolver valores de tipo básico o de tipo complejo (por ejemplo, registros, vectores, conjuntos o incluso tablas), permitiéndose tipificación polimórfica para funciones abstractas o genéricas (referencia a variables de tipo objeto). Con PL/pgSQL puede usar todos los tipos de datos, columnas, operadores y funciones de SQL. Debido a que las funciones PL/pgSQL corren dentro de PostgreSQL, estas funciones funcionarán en cualquier plataforma donde PostgreSQL corra [PostgreSQL 2003].

1.7.3 PgAdmin 1.14.2

Es una potente plataforma de administración y desarrollo para la base de datos PostgreSQL, libre para cualquier uso. PgAdmin es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usan en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL sencillas a complejas para el desarrollo de bases de datos [pgAdmin 2010].

Conclusiones Parciales

- En el capítulo quedó definido el proceso de limpieza de datos como una serie de pasos que permiten detectar errores en los datos y eliminar los mismos para obtener datos de alta calidad, ofreciendo con esto una mejora en los procesos de toma de decisiones de las organizaciones.

- Se han propuesto métodos y algoritmos tales como la distancia de edición y la similitud de jaro, con sus técnicas para eliminar las anomalías existentes en las bases de datos del GESPRO, centrándose en las irregularidades y duplicados ya que a las anomalías de alcance, se le dan solución por el equipo del laboratorio de gestión de proyecto.

- A pesar de que las herramientas mencionadas; implementan diferentes algoritmos de limpieza de datos; se concluye que el uso de una de estas en la BD del GESPRO conllevaría a la obligatoria capacitación del personal para la configuración y ejecución de la misma para lo que el personal del GESPRO no dispone de tiempo ni recursos.

- Se tomará como guía los principales pasos de la metodología de limpieza de datos propuesta por Leslie M. Tierstein para una buena comprensión, planeación, preparación y solución a la situación problémica.

- Se implementarán funciones SQL en el lenguaje y plataforma propuestos para lograr que el cliente final pueda interactuar de forma práctica con los algoritmos escogidos.

CAPÍTULO 2 PROCESO DE LIMPIEZA DE DATOS

En este capítulo se desarrollará paso a paso la metodología propuesta, donde quedarán definidos e implementados los algoritmos de limpieza de datos que permitan mejorar la calidad de los datos de las bases primarias del GESPRO.

2.1 Planeación y Preparación

En esta fase de planeación y preparación se describirá el proceso de limpieza de datos, teniendo en cuenta las características de los datos a los cuales se les realizará la limpieza y la identificación de los errores que se presentan. Se determinan los antecedentes que originaron la situación por la cual se presentan estos errores en la BD del GESPRO y luego se definen los objetivos que se quieren alcanzar.

2.1.1 Contextualización y compresión del negocio

La gestión de proyecto se encarga de organizar y administrar todos los recursos necesarios para un proyecto de manera eficiente, permitiendo que este se termine dentro del alcance, tiempo y con los costos definidos inicialmente. Actualmente a nivel internacional, existe una tendencia a utilizar herramientas informáticas para apoyar la gestión de proyectos y facilitar así el trabajo de los especialistas [Cabrera Lamadrid and Cadenas del Llano 2012].

En la UCI, con el fin de promover la formación de los estudiantes desde su participación en proyectos reales, existe una red de centros de producción en diferentes zonas del país donde se organizan desarrolladores, que colaboran y desarrollan un conjunto de soluciones informáticas bajo una estrategia única que se gestiona desde la sede central. Teniendo en cuenta además el tamaño de la organización (6000 usuarios que contribuyen al desarrollo de proyectos de software) y el volumen de datos que se manejan, se identificó como una necesidad la introducción de herramientas para la ayuda a la toma de decisiones a diferentes niveles: nivel de persona, nivel de proyecto, nivel de centro de producción, nivel alta gerencia. [Piñero Pérez et al. 2011] Para dar respuesta a esta necesidad la UCI desarrolló un Paquete de Gestión de Proyecto (GESPRO).

GESPRO cuenta con diversas funcionalidades para facilitar la gestión centralizada de todos los proyectos de la UCI a distintos niveles [Sánchez Giraut 2012]:

- Esta herramienta permite el control de la producción de la universidad organizado por proyectos en cuatro niveles fundamentales: el nivel de las personas, el nivel de los proyectos, el nivel de los centros de desarrollo y el nivel gerencial de la universidad.

- Este entorno permite la integración con el nuevo modelo de formación integrado formación-producción-investigación.

- El sistema está alineado con los avances tecnológicos en la gestión de proyectos, permite la dirección integrada de proyectos a diferentes niveles, nivel de personas, proyectos, centros y organizaciones. Además de ser competitivo con los mejores software de gestión de proyectos del mundo.

- Los componentes del entorno son desarrollados o dominados por la red de Centros de Desarrollo de la Universidad y basados en tecnologías libres como requisito para la soberanía tecnológica y la seguridad.

- Este entorno debe permitir su integración con otros sistemas.

Este sistema ha sido desarrollado utilizando los modelos de líneas de productos de software como modelo industrial de desarrollo. En la actualidad GESPRO en su versión 11.05 es una plataforma extensible que está formada a partir de la integración de más de 18 herramientas libres, comercializadas bajo licencia GPL [Martínez Vigil 2012]. El Paquete de Gestión de Proyectos GESPRO v1.0 se encuentra registrado en el Centro Nacional de Derecho de Autor (CENDA) con No. Registro 1540-2010 [Piñero Pérez, et al. 2011].

Posee funcionalidades que permiten: la gestión de portafolios de proyectos, la gestión del alcance de productos, la gestión del tiempo, la gestión de riesgos de proyectos, la gestión de comunicaciones, la gestión de recursos humanos y materiales, el monitoreo y control de la plataforma, el control de versiones y la gestión documental. La herramienta GESPRO está formada por cuatro grupos principales de herramienta, las cuales se mencionan a continuación [Sánchez Giraut 2012]:

- Herramientas para la dirección integrada de proyectos.

- Herramientas para la gestión documental y el control de versiones.

- Herramientas para el monitoreo, la administración y la recuperación ante fallos.

- Herramientas para el trabajo colaborativo y la ayuda a la toma de decisiones.

Entre las herramientas que lo componen se encuentran [Cabrera Lamadrid and Cadenas del Llano 2012]:

Redmine v0.9.3, PATDSI Generador de Reportes v1.0, eXcriba v1.0, SVN, Statsvn, SVNPlot, VMWare (Virtual Center), CAS, Zimbra, Afresco y Suite Pentaho.

En las siguientes imágenes se muestran los módulos que los componen.

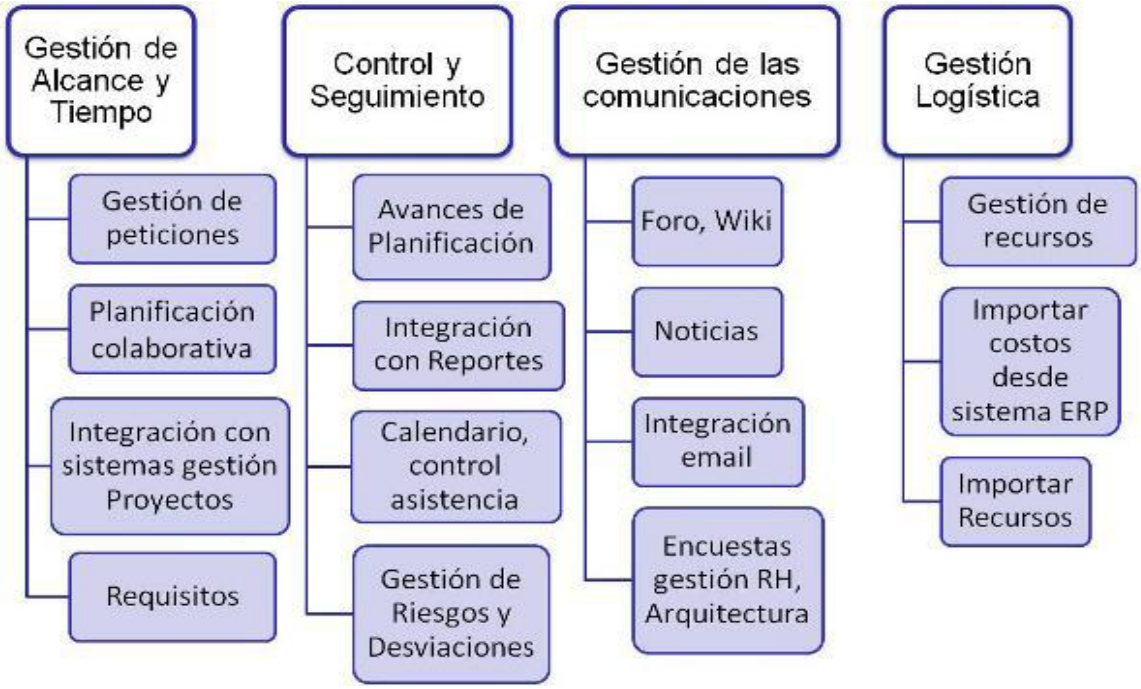


Figura 2. Módulos del GESPRO.

Figura 3. Módulos del GESPRO.

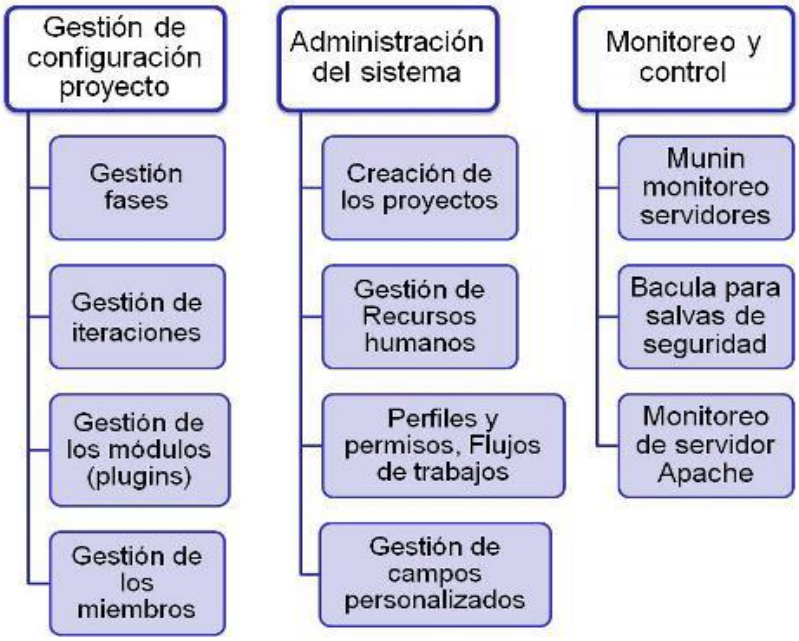


Figura 4. Módulos del GESPRO.

GESPRO se ha aplicado con buenos resultados en la red de centros de la UCI, que incluye 13 centros de desarrollo de software ubicados en la sede central de la UCI y cinco centros regionales. Actualmente es

utilizado por más de 6000 usuarios que gestionan actividades de más de 150 proyectos entre los que se encuentran proyectos para la informatización nacional como para la exportación[Martínez Vigil 2012].

Esta herramienta de gestión de proyectos, por los grandes volúmenes de datos que maneja en sus BD contiene innumerable información. Dicha información es utilizada para el proceso de toma de decisiones, por lo cual deben contar con una alta confiabilidad. Para esto es necesario realizar una limpieza de datos para corregir cualquier anomalía presente en las BD. Esta necesidad surge a partir de que se plantea que si no se limpian los datos, el conocimiento generado a partir de estos datos para la toma de decisiones no sería completamente correcto, debido a que estaría basado en datos que no tienen la calidad requerida.

Una de las principales razones por lo que estos datos se encuentran sucios se basan en la falta de un modelo relacional de BD, el cual garantizaría herramientas para evitar la duplicidad de registros, a través de campos claves o llaves, dicho modelo también sería capaz de aportar integridad referencial, así, al eliminar un registro eliminaría todos los registros relacionados dependientes, además que este favorecería la normalización por ser más comprensible y aplicable. Dicha BD se encuentra mostrando un conjunto de datos planos relacionando únicamente una tabla con otra mediante funciones y campos como el ID.

2.2 Análisis y diseño conceptual

Trabajar bajo el análisis conceptual de una situación, se refiere a la abstracción de hechos reales de los cuales se emite un concepto o es posible expresar una idea acerca de ellos. Para poder realizar dicha abstracción en el área específica del GESPRO fue necesario tener los requerimientos formulados por los usuarios con respecto a los problemas a tratar, o sea conocer los errores presentados en la BD, dígase irregularidades o duplicados, para con esto tener un perfil sobre la limpieza de datos a realizar.

2.2.1 Determinación de una muestra de los datos para identificar los principales problemas

Dada la alta complejidad que puede llegar a requerir un análisis exhaustivo sobre la calidad de los datos de la BD del GESPRO debido al gran volumen de datos que se guardan en el mismo, y la información sensible que se maneja, se seleccionó un subconjunto de esta como muestra para analizar de acuerdo con la taxonomía seleccionada, las anomalías encontradas en la misma.

Se tomaron las tuplas de la tabla peticiones (ISSUES) correspondientes al centro CEIGE, por ser estas las más usadas, con más acceso y propensa a entrada de datos erróneos, por el uso intensivo de profesores

y estudiantes de la UCI que acceden al GESPRO, además de que es una de las tablas que más evidencias aporta para la base de casos. La tabla comprende 16 columnas, de estas, la columna *tarea* es la que permite la entrada de texto, siendo así la expuesta a la entrada de datos que contienen errores. En la muestra seleccionada dicha tabla está compuesta por 10283 peticiones.

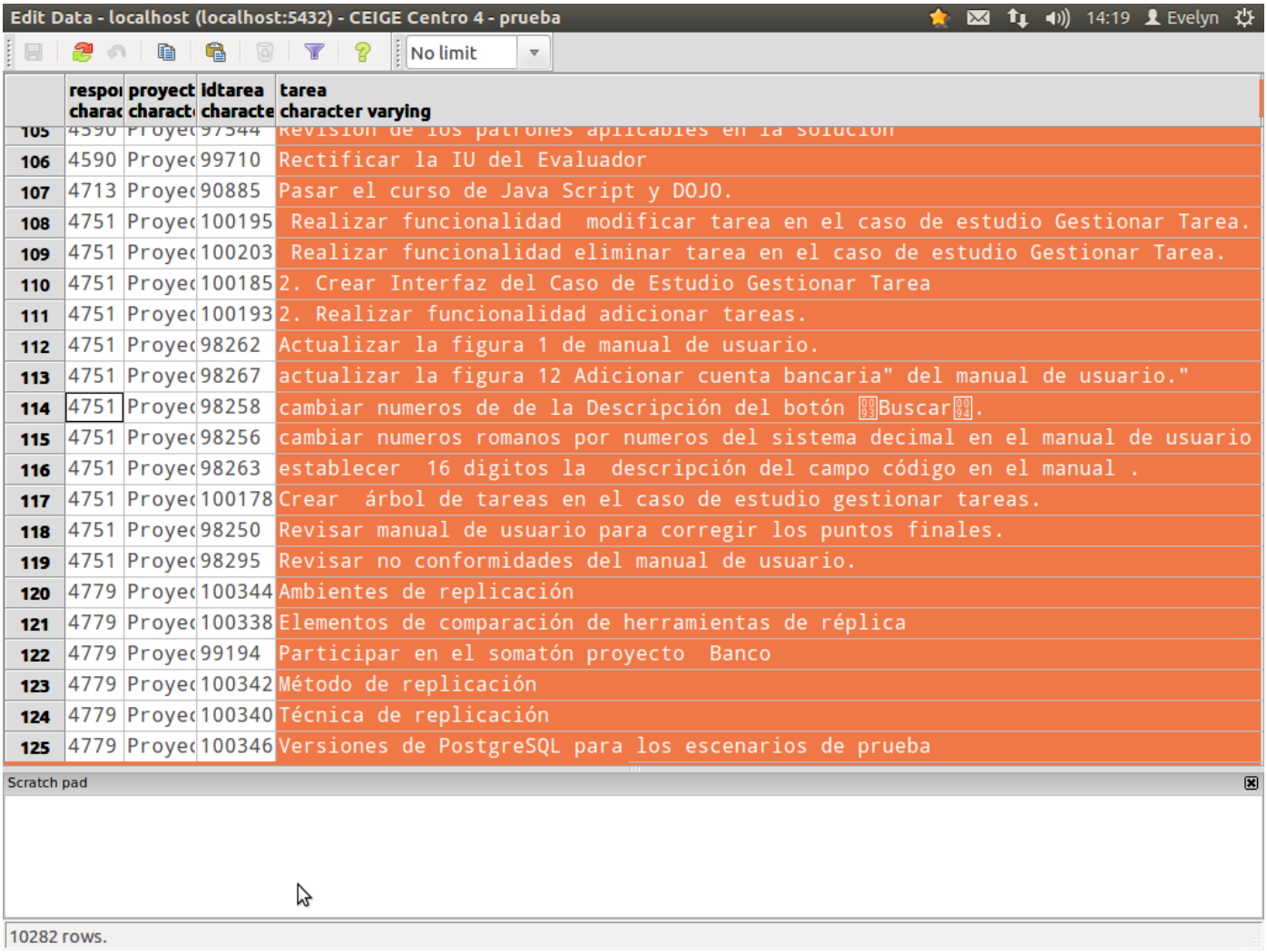


Figura 5: Atributo tarea perteneciente a la tabla ISSUES

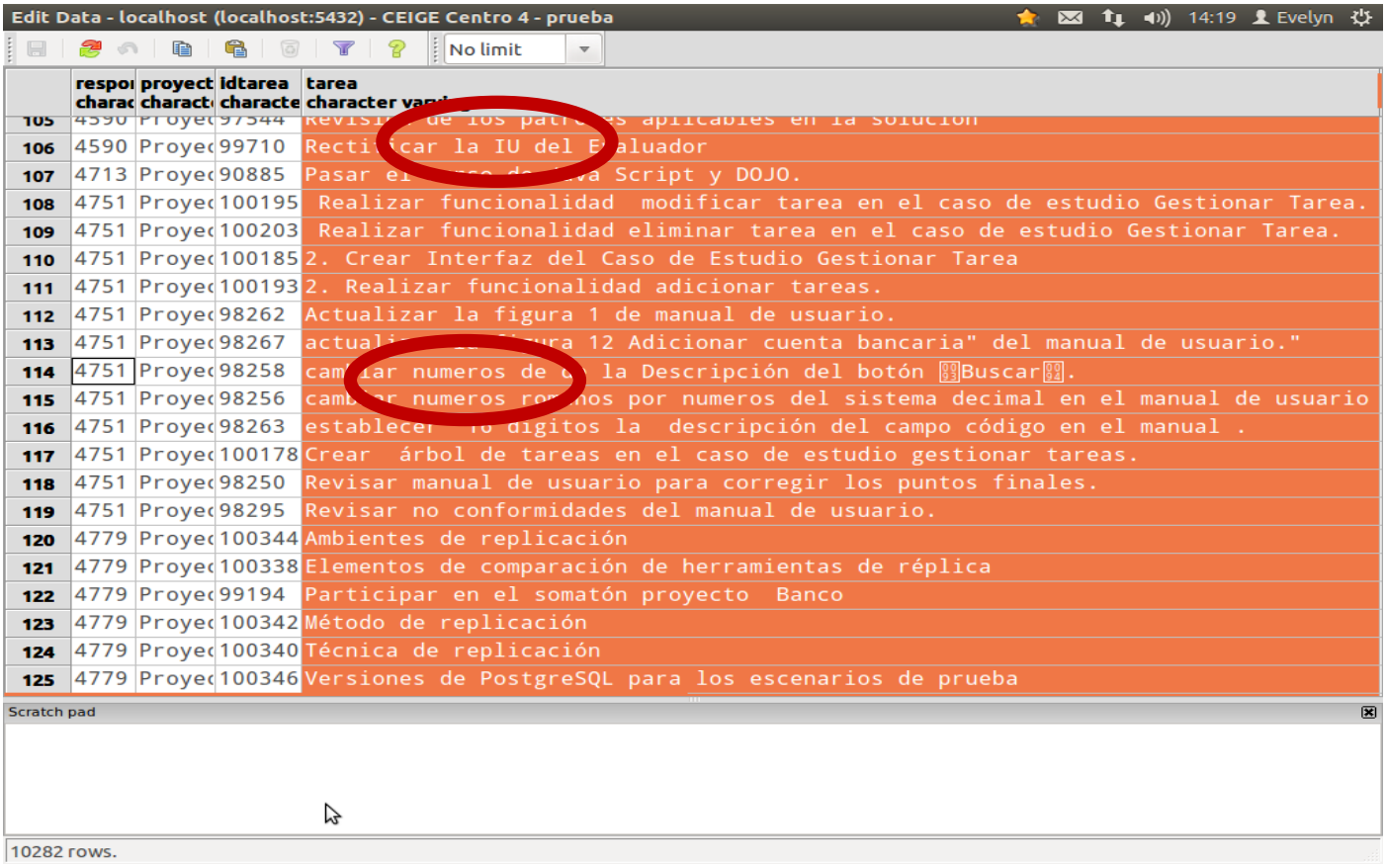
2.2.2 Identificación de los problemas en la muestra con base en una taxonomía de problemas comunes.

Se realizó una revisión tupla a tupla de la muestra seleccionada donde se identificaron los siguientes problemas:

Tabla 4: Anomalías identificadas en la muestra.

Tipo de anomalía	Cantidad presentada en la muestra
Irregularidades	124
Duplicados	547
No estandarización de cadenas	26
Errores ortográficos	1436

Sobre los problemas en la base de datos del GESPRO tienen predominio las irregularidades, pertenecientes a la anomalía de errores sintácticos ya que en ella están inmersos los errores relacionados con las abreviaturas, adición u omisión de letras debido a la cercanía de estas en el teclado, no estandarización de cadenas, (ejemplo el uso de IU o interfaz de usuario indistintamente) además de errores ortográficos(terminación ción, las palabras terminadas en -aje se escriben con j, se escriben con s las palabras que terminan en ersa, erse, erso, las palabras que empiezan con hie- y hue- hui-, hia- llevan h, entre otras reglas definidas); y en la anomalía de errores semánticos los problemas de duplicados (cuando dos o más tuplas reflejan una misma entidad). Las mismas pueden observarse en las Figuras 6 y 7.



	respor charac	proyect charact	idtarea characte	tarea character vari
105	4590	Proyec	97544	Revisi de los patr s aplicables en la solucion
106	4590	Proyec	99710	Recti car la IU del E aluador
107	4713	Proyec	90885	Pasar el va Script y DOJO.
108	4751	Proyec	100195	Realizar funcionalidad modificar tarea en el caso de estudio Gestionar Tarea.
109	4751	Proyec	100203	Realizar funcionalidad eliminar tarea en el caso de estudio Gestionar Tarea.
110	4751	Proyec	100185	2. Crear Interfaz del Caso de Estudio Gestionar Tarea
111	4751	Proyec	100193	2. Realizar funcionalidad adicionar tareas.
112	4751	Proyec	98262	Actualizar la figura 1 de manual de usuario.
113	4751	Proyec	98267	actuali ura 12 Adicionar cuenta bancaria" del manual de usuario."
114	4751	Proyec	98258	cam iar numeros de l la Descripción del botón Buscar.
115	4751	Proyec	98256	camb r numeros ro nos por numeros del sistema decimal en el manual de usuario
116	4751	Proyec	98263	establece 10 digitos la descripción del campo código en el manual .
117	4751	Proyec	100178	Crear árbol de tareas en el caso de estudio gestionar tareas.
118	4751	Proyec	98250	Revisar manual de usuario para corregir los puntos finales.
119	4751	Proyec	98295	Revisar no conformidades del manual de usuario.
120	4779	Proyec	100344	Ambientes de replicación
121	4779	Proyec	100338	Elementos de comparación de herramientas de réplica
122	4779	Proyec	99194	Participar en el somatón proyecto Banco
123	4779	Proyec	100342	Método de replicación
124	4779	Proyec	100340	Técnica de replicación
125	4779	Proyec	100346	Versiones de PostgreSQL para los escenarios de prueba

Figura 6: Anomalías referentes a la no estandarización de cadenas y errores ortográficos

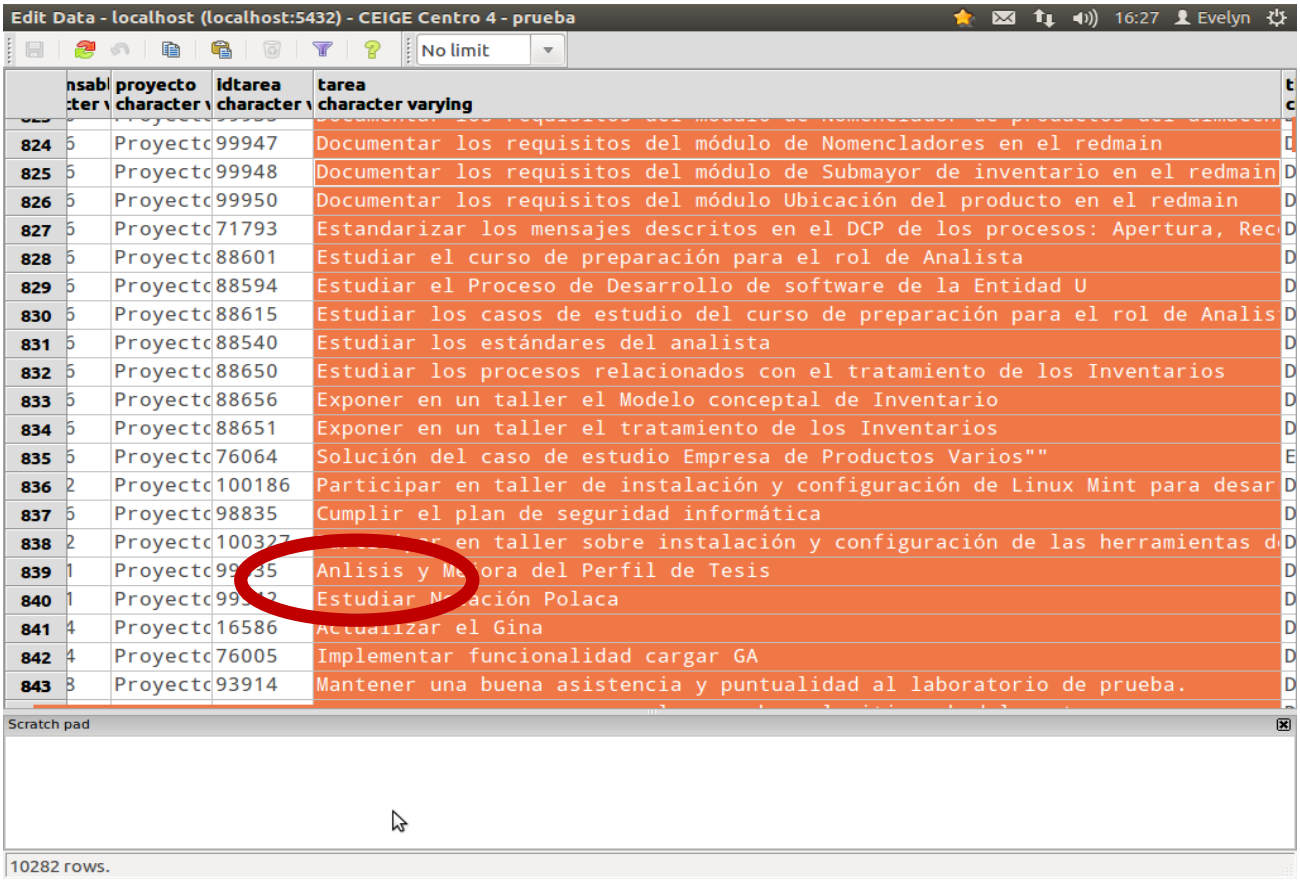


Figura 7: Anomalías referentes a las irregularidades

2.2.3 Realización de listado de tablas a limpiar

Dentro de las áreas de gestión de proyectos que abarca el GESPRO se encuentran la gestión del alcance, tiempo y la calidad, en ellas se refleja las peticiones que se manejan, refiriéndose al dominio de soluciones de gestión de proyectos y a la línea de productos de software que lo representa.

Para manipular estos datos el GESPRO interactúa con la tabla ISSUES, la cual posee como atributo TAREA de tipo de dato TEXT. Este campo brinda una breve descripción de las actividades a realizar, las cuales están en función del alcance y objetivos del proyecto. Estos son datos de importancia para los equipos de proyectos, ya que además de las áreas mencionadas anteriormente también inciden de manera directa en la gestión de los recursos humanos, siendo estas actividades evidencias del trabajo realizado por cada persona del equipo. El que existan errores en esta información influye en que las

evidencias recopiladas en ella tengan valores que no son correctos, por tanto la futura inferencia de conocimiento sobre estas evidencias no serían correctas.

Por lo antes planteado se decide realizar la limpieza de datos a la tabla ISSUES (peticiones) de la BD del GESPRO correspondiente al centro CEIGE, específicamente la columna tarea. Se observa que la entrada de datos para los demás valores de la tabla se hace de forma restringida, o sea que ya en la herramienta están definidos los valores que pueden tomar cada uno de los campos, siendo solo el nombre de las tareas lo que queda a responsabilidad del usuario y donde además se cometen la mayor cantidad de errores. Con esto se podrá contar con la calidad en los datos necesaria para la construcción de la base de conocimientos.

2.3 Realizar la limpieza

Al realizar la limpieza se tendrán definidos los métodos a utilizar, los cuales serán implementados mediante funciones en el lenguaje PL/pgSQL de Postgres. El cliente necesita tener una consulta que pueda ser corrida en las bases de datos en momentos que la utilización de los sistemas sea mínima, de esta forma se mejora la calidad del dato, no afecta la utilización de la herramienta y no utiliza recursos para que se realice el proceso, el mismo se haría de manera automática. Al finalizar se emitirá un informe donde se realizarán comparaciones de la base de datos antes y después de la limpieza de datos.

2.3.1 Definición de métodos de limpieza para los problemas seleccionados

Se diseñarán los algoritmos usando funciones SQL a partir de las funciones de similitud de Jaro y distancia de edición. Estos métodos proponen realizar la detección y corrección de duplicados, además de la corrección de errores ortográficos, no estandarización de cadenas e irregularidades respectivamente. Estos métodos se definen como se muestra a continuación:

- **Distancia de Edición**

En teoría de la información y ciencias de la computación se llama Distancia de Levenshtein, distancia de edición, o distancia entre palabras, al número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra[Cáceres 2008]. Se entiende por operación, bien una inserción, eliminación o la sustitución de un carácter. Es útil en programas que determinan cuán similares son dos cadenas de caracteres, como es el caso de los correctores de ortografía.

En el modelo original, cada operación de edición tiene costo unitario, siendo referido como distancia de Levenshtein. En [Needleman and Wunsh 1970]se modificó para permitir operaciones de edición con distinto costo, permitiendo modelar errores ortográficos y tipográficos comunes. Es usual encontrar "n" en lugar de "m" (o viceversa) entonces, tiene sentido asignar un costo de sustitución menor a este par de caracteres que a otros dos sin relación alguna. Por otro lado, [Lowrence and Wagner 1975]introducen un modelo que permite la trasposición de dos caracteres adyacentes como una cuarta operación de edición, usualmente referido como distancia de Damerau-Levenstein.

Los modelos anteriores tienen una desventaja: la distancia entre dos cadenas de texto carece de algún tipo de normalización, lo cual los hace imprecisos. Por ejemplo, tres errores son más significativos entre dos cadenas de longitud 4 que entre dos cadenas de longitud 20. Existen varias técnicas de normalización. Las más simples lo hacen dividiendo por la longitud de la cadena más larga o por la suma de la longitud de ambas cadenas. La investigación de [Marzal and Vidal 1993]propone dividir por el número de operaciones de edición. Más recientemente, [Yujian and Bo 2007]desarrollaron la primera técnica de normalización que satisface la desigualdad triangular. Por otro lado, [Ristad and Yianilos 1998]lograron un modelo que aprende automáticamente los costos más óptimos de las operaciones de edición a partir de un conjunto de datos de entrenamiento.

La distancia de edición no normalizada puede ser calculada en O(nm) mediante el algoritmo de programación dinámica propuesto por[Wagner and Fischer 1974]. Mientras que [Masek 1980]presenta un algoritmo que toma O(n·max(1,m/log n)) siempre que los costos de las operaciones de edición sean múltiplos de un único real positivo y que sea finito. Además, [Hyyrö 2012]presenta un algoritmo basado en operaciones a nivel de bits para verificar si la distancia de Damerau-Levenshtein entre dos strings es menor que cierta constante d, el cual toma O(|Σ| + [d / w]· m), donde w es el tamaño de la palabra definida por el procesador. La técnica de normalización propuesta por [Marzal and Vidal 1993]toma $O(n^2m)$, siendo reducida a O(nm log m) y a O(nm) mediante programación fraccionaria. Por otro lado,[Eğecioğlu and Ibel 1996]proponen algoritmos paralelos eficientes. El orden computacional de la técnica de normalización de [Yujian and Bo 2007]es igual al del algoritmo que se utilice para calcular la distancia no normalizada, al ser función directa de esta última.

A continuación se indica el pseudocódigo de dicha función:

```
Int LevenshteinDistance(char str1[1..lenStr1], char str2[1..lenStr2])

// d es una tabla con lenStr1+1 filas y lenStr2+1 columnas

Declare int d[0..lenStr1, 0..lenStr2]

// i y j se utiliza para iterar sobre str1 y str2

Declare int i, j, cost

for i from 0 to lenStr1

d[i, 0] := i

for j from 0 to lenStr2

d[0, j] := j

for i from 1 to lenStr1

for j from 1 to lenStr2

if str1[i] = str2[j] then cost := 0

else cost := 1

d[i, j] := minimum(

d[i-1, j] + 1,     // supresión

d[i, j-1] + 1,     // inserción

d[i-1, j-1] + cost   // sustitución

                        )

return d[lenStr1, lenStr2]
```

- **Similitud de Jaro**

Esta métrica computa los caracteres comunes tales que el *i*-ésimo carácter es igual al *j*-ésimo carácter donde el valor absoluto de *i-j* sea menor o igual que la mitad del mínimo entre las longitudes de los dos textos [11]. Además se calcula también, el número de transposiciones de estos caracteres comunes como la comparación fallida del *i*-ésimo de estos entre las dos diferentes cadenas.

La distancia de Jaro Dj de dos cadenas dadas S1 y S2 es:

$$d_j = \begin{cases} 0 & si\ m=0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & otro\ caso \end{cases}$$

Donde

m: número de caracteres que coincida.

t: número medio de transposiciones.

Dos caracteres de S1 y S2, respectivamente, se consideran coincidentes sólo si son el mismo y no más allá de:

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1 \leq 1$$

Cada carácter de S1 se compara con todos los caracteres que coincida en S2. El número de caracteres coincidentes (donde el orden de la secuencia es diferente) dividido por 2 define el número de transposiciones. Por ejemplo. En comparación CRATE con TRACE, solo R A E son los caracteres que coinciden, es decir m=3. Aunque 'CT' aparece en ambas cadenas, están más alejados que 1, es decir, (5/2) -1 = 1,5. Por lo tanto, t = 0. En Dwane contra Duane las letras coincidentes ya están en el mismo orden, DANE, por lo que no son necesarias transposiciones.

La distancia Jaro-Winkler utiliza una escala p prefijo que da puntuaciones más favorables para las cadenas que coinciden desde el principio para un conjunto l longitud del prefijo. Dadas dos cadenas S1 y S2, su Jaro-Winkler distancia Dw es la siguiente:

$$d_w = d_j + (\ell p(1 - d_j))$$

Dónde:

dj: distancia Jaro para las cadenas S1 y S2

l: longitud del prefijo común en el principio de la cadena, hasta un máximo de 4 caracteres

p: es un factor de escala constante para la cantidad de la cuenta se ajusta al alza para tener prefijos comunes. p no debe exceder de 0,25, de lo contrario la distancia puede ser mayor que 1. El valor estándar para esta constante en la obra de Winkler es p = 0,1. La similitud de Jaro puede ser calculada en O(n).

2.3.2 Limpieza de los datos.

La limpieza de datos es efectuada en dos tiempos. En un primer tiempo se aplica la función de similitud de Jaro en busca de duplicados. Luego de detectados y corregidos los duplicados se prosigue con la aplicación de la función de distancia de edición para la corrección de errores ortográficos, no estandarización de cadenas e irregularidades. En ambos casos el procedimiento depende de la entrada de las tareas previstas para realizar la limpieza.

En el caso del procedimiento llevado a cabo para la similitud de Jaro son usadas varias funciones auxiliares encargadas de calcular el índice de similitud, la cual calcula un índice para saber cuan similar es una palabra de otra. Dividir la tarea, pasándole dos cadenas y un idtarea, compara cada palabra de la tarea con la otra tupla luego de calculado el índice de similitud. Además de usar una función para convertir la tarea ya dividida en arreglo, usando delimitadores para su mejor tratamiento. En las figuras a continuación se pueden observar fragmentos de la implementación de dichas funciones.

```
if s1 > s2 then
        t := (s1 / 2) - 1;
else
        t := (s2 / 2) - 1;
end if;
```

Figura 8: Cálculo de las transposiciones

```
return (1.0 / 3.0) * ( (m / s1) + (m / s2) + ((m - t) / m));
```

Figura 9: Cálculo del índice de similitud

```
if promedio > 0.85 then
        delete from prueba where prueba.idtarea = $3;
        eliminados := array_append(eliminados, $3);
end if;

return eliminados;
```

Figura 10: Comparación del índice de similitud para detectar duplicados

```
FOR i IN array_lower(elems, 1) .. array_upper(elems, 1) LOOP
    IF char_length(elems[i]) > 3  THEN
       palabra := rtrim(elems[i],'.');
       palabra := rtrim(palabra,',');
       palabra := rtrim(palabra,';');
       palabra := rtrim(palabra,':');
       resp := array_append(resp, palabra);
    END IF;
END LOOP;
```

Figura 11: Función para eliminar caracteres extraños o no deseados

```
for oracion1 in select * from prueba order by idtarea asc limit 10 loop
       limpio1 := string_to_rows(oracion1.tarea);
       for oracion2 in Select * from prueba where prueba.idtarea > oracion1.idtarea order by idtarea asc limit 500 loop
              limpio2 := string_to_rows(oracion2.tarea);
              rest := array_cat(rest, dividirOracion(limpio1, limpio2, oracion2.idtarea));
       end loop;
end loop;
return rest ;
```

Figura 12: Función principal para la eliminación de duplicados

Estas tareas son ordenadas por un idtarea de forma ascendente para realizar la comparación tupla a tupla, de forma que nunca se llegue a comparar una tupla con ella misma, de la siguiente forma:

```
Select * from prueba where prueba.idtarea > oracion1.idtarea order by idtarea asc
```

Finalmente en una función principal se realiza la detección y eliminación de cualquier tupla que se haya detectado como duplicada.

Por otra parte la función de distancia de edición se encarga de tomar las tareas en una variable. Elimina cualquier carácter extraño contenido en la tupla correspondiente a la tarea para que solo queden palabras. Después de limpia de caracteres no deseados, la variable es convertida en arreglo para tratarla y aplicarles las reglas ortográficas correspondientes.

```
for var in select idtarea, tarea from prueba
loop
  temporal:= (select regexp_replace(var.tarea, '[^a-zA-Z0-9ñÑáéíóúüÁÉÍÓÚ]+','/','g'));
  temporal:= (select ltrim(temporal,'0123456789./ '));
  temporal:= (select lower(temporal));
  arreglo := (select regexp_split_to_array(temporal, '[^a-zA-Z0-9ñÑáéíóúüÁÉÍÓÚ]+'));
fin:= (select array_length(arreglo,1))::integer;
for i in 1..fin
loop
palabra_ant:=arreglo[i-1];
palabra:= arreglo[i];
palabra_sgt:=arreglo[i+1];

--ABREVIATURAS A PARTIR DE AQUI--
if palabra='nc'then remplazada:= replace(palabra,'nc','no conformidad');
aux[i]:=remplazada;
elsif (palabra='nc' and palabra_ant='las') then remplazada:= replace(palabra,'nc','no conformidades');
aux[i]:=remplazada;
elsif palabra='neg' then remplazada:= replace(palabra,'neg','negocio');
aux[i]:=remplazada;
```

Figura 13: Eliminación de caracteres extraños y abreviaturas

```
--ERRORES ORTOGRAFICOS A PARTIR DE AQUI--
--Las palabras terminadas en ión se tildan
elsif right(palabra,3)='ion' then remplazada:= replace(palabra,'ion','ión');
aux[i]:=remplazada;
--Las palabras terminadas en -aje se escriben con j
elsif right(palabra,3)='age' then remplazada:= replace(palabra,'age','aje');
aux[i]:=remplazada;
--Las palabras terminadas en -voro y-vora se escriben con v.(excepto víbora)
elsif right(palabra,4)='boro' then remplazada:= replace(palabra,'boro','voro');
aux[i]:=remplazada;
elsif right(palabra,4)='bora' then remplazada:= replace(palabra,'bora','vora');
aux[i]:=remplazada;
elsif palabra='vívora' then remplazada:= replace(palabra,'vívora','víbora');
aux[i]:=remplazada;
--Las palabras terminadas en -ésimo y -ésima se escriben con s. (excepto décimo y décima)
elsif right(palabra,5)='écimo' then remplazada:= replace(palabra,'écimo','ésimo');
aux[i]:=remplazada;
elsif right(palabra,5)='écima' then remplazada:= replace(palabra,'écima','ésima');
aux[i]:=remplazada;
elsif palabra='désimo' then remplazada:= replace(palabra,'désimo','décimo');
aux[i]:=remplazada;
elsif palabra='désima' then remplazada:= replace(palabra,'désima','décima');
aux[i]:=remplazada;
-- Las palabras terminadas en -gente -gencia -gio se escriben con g.
elsif right(palabra,5)='jente' then remplazada:= replace(palabra,'jente','gente');
aux[i]:=remplazada;
elsif right(palabra,6)='jencia' then remplazada:= replace(palabra,'jencia','gencia');
```

Figura 14: Aplicación de reglas ortográficas genéricas

```
--IRREGULARIDADES A PARTIR DE AQUI--
elsif palabra='almacen' then remplazada:= replace(palabra,'almacen','almacén');
aux[i]:=remplazada;
elsif palabra='anlisis' then remplazada:= replace(palabra,'anlisis','análisis');
aux[i]:=remplazada;
elsif palabra='alos' then remplazada:= replace(palabra,'alos','a los');
aux[i]:=remplazada;
elsif palabra='arquitectonicos' then remplazada:= replace(palabra,'arquitectonicos','arquitectónicos');
aux[i]:=remplazada;
elsif palabra='areas' then remplazada:= replace(palabra,'areas','áreas');
aux[i]:=remplazada;
elsif palabra='boton' then remplazada:= replace(palabra,'boton','botón');
aux[i]:=remplazada;
elsif palabra='basicos' then remplazada:= replace(palabra,'basicos','básicos');
aux[i]:=remplazada;
elsif palabra='categoria' then remplazada:= replace(palabra,'categoria','categoría');
aux[i]:=remplazada;
elsif palabra='catalogos' then remplazada:= replace(palabra,'catalogos','catálogos');
aux[i]:=remplazada;
elsif palabra='consultoria' then remplazada:= replace(palabra,'consultoria','consultoría');
aux[i]:=remplazada;
elsif palabra='calses' then remplazada:= replace(palabra,'calses','clases');
aux[i]:=remplazada;
elsif palabra='digito' then remplazada:= replace(palabra,'digito','dígito');
aux[i]:=remplazada;
```

Figura 15: Anomalías referentes a las irregularidades

Conclusiones Parciales

- El establecimiento de una taxonomía de errores comunes permitió la implementación de algoritmos necesarios para realizar el proceso de limpieza de datos.

- Las distancias de edición y jaro, favorecen la solución al problema de estandarizar cadenas, corregir irregularidades, errores ortográficos y permiten la detección de duplicados.

- Se desarrollaron los primeros pasos de la metodología propuesta, quedando así implementados los métodos de limpieza de datos necesarios para corregir las anomalías presentadas en la BD del GESPRO.

CAPÍTULO 3 VALIDACIÓN DE LA SOLUCIÓN

Evaluar y validar que la solución desarrollada y las afirmaciones referentes a la solución sean aceptables para la comunidad investigadora es un paso fundamental de cualquier investigación. En este capítulo se llevó a cabo el análisis del proceso de validación de la solución propuesta. Mostrando indicadores, métodos de validación y complejidad de los algoritmos desarrollados.

3.1 Indicadores para medir la eficiencia de los algoritmos.

La tarea de la programación consiste, entre otras cosas, en la confección de un algoritmo o programa que resuelva satisfactoriamente el problema planteado. Es evidente que la corrección es la condición indispensable que debe satisfacer un algoritmo, es decir, que haga exactamente lo que se espera de él. Existen una serie de características adicionales que permiten establecer el nivel de calidad del algoritmo: legibilidad, modificabilidad, modularidad, reusabilidad, portabilidad, etc. Pero para que sea definitivamente satisfactorio conviene que resuelva el problema lo más rápido posible o que use la menor cantidad de memoria posible y, mejor aún, ambas cosas a la vez. Por tanto es necesario poder medir el consumo de recursos, tiempo y memoria, de un programa para establecer su nivel de calidad. Además esta información será útil para efectuar comparaciones entre diferentes alternativas que resuelvan el mismo problema y permitirá seleccionar la mejor opción, aquella que menos recursos consuma[Abad 2011]. En este siguiente epígrafe se estará analizando la eficiencia de los algoritmos.

Los indicadores de eficiencia miden el nivel de ejecución del proceso y el rendimiento de los recursos utilizados por un proceso. Existen un conjunto de indicadores para la medición de eficiencia en los algoritmos, tales como:

- Tiempo de ejecución.
- Efectividad.
- Volumen de datos.
- Complejidad.

- Claridad.
- Memoria usada

En este caso específico se proponen utilizar los siguientes:

Tiempo de ejecución: Tiempo que demora la ejecución de los algoritmos propuestos.

Depende de 2 factores fundamentales:

- El tamaño de los datos de entrada.
- El contenido de los datos de entrada.

Volumen de datos: Cantidad de los datos a evaluar, en este caso se hace referencia a la cantidad de tuplas existentes como tareas en la BD.

Efectividad: Con este indicador se mide la variable calidad de datos. Es la relación entre los resultados logrados y los resultados propuestos, o sea permite medir el grado de cumplimiento de los objetivos planificados, en este caso la reducción de los errores o anomalías en los datos.

3.2 Propuesta del experimento para validar la investigación

Para realizar una propuesta del experimento para la validación de la investigación se llevó a cabo el análisis de varios métodos de validación investigados. Los siguientes métodos proporcionan una vía para realizar la evaluación y validación de una solución desarrollada [Vaishnavi and William Kuechler 2007]:

- **Experimentación**

La experimentación es utilizada para validar o rechazar un conjunto de hipótesis relacionadas con las afirmaciones acerca de la solución. Estas hipótesis no pueden ser probadas matemática o lógicamente, por lo que es necesario generar un conjunto de datos del sistema y luego utilizar esta información para validar o rechazar las hipótesis. La experimentación ayudará a establecer resultados asociados con la solución del problema de investigación en situaciones donde la recogida y análisis de datos es el único método factible de validación.

- **Demostración**

Se aplica desarrollando una o varias situaciones particulares de un problema y trabaja para ese conjunto de situaciones predefinidas. El método es especialmente relevante cuando la demostración de una solución en sí misma se considera una contribución, consta de dos etapas fundamentales: la construcción de la respuesta que permite afirmar que esta es realizable y la demostración de que la misma es razonable para un conjunto de situaciones predefinidas. Como resultado la demostración puede exponer las deficiencias de la solución o por el contrario que es viable y aceptable. Las pruebas exhaustivas aumentan la confianza en las conclusiones arribadas, si las situaciones de prueba están diseñadas apropiadamente, entonces la construcción de la respuesta y sus pruebas para estas situaciones puede demostrar la validez de la misma.

- **Simulación**

La simulación es usada para evaluar y validar una solución a un problema complejo tal que la misma no pueda ser demostrada matemáticamente como válida. La evaluación y validación de esta en el ámbito de la vida real es poco viable y costoso, entonces el problema y su respuesta deben ser modelados con precisión en una computadora. Para la realización de la simulación es necesario contar con el modelo conceptual del problema y su solución para que sean simulados en una computadora y un conjunto de datos de prueba iniciales. Este método ofrece una forma razonable y rentable de evaluación y validación de un resultado y brinda la alternativa de poner a prueba este en la vida real lo que puede ser a la vez costoso y consume mucho tiempo, o tal vez ni siquiera sea factible.

- **Marcadores**

Los marcadores son utilizados para demostrar que una solución tiene un rendimiento razonable o es mejor que alguna otra disponible. Es usado generalmente cuando no hay métricas disponibles para medir el rendimiento de la misma y se hace necesario probar que esta respuesta desarrollada es superior a otras. Si no existen marcadores disponibles para validarla es efectivo crear un escenario o varias clases de escenarios para evaluarla y demostrar su superioridad a otras soluciones disponibles. Para la aplicación de este método es necesario identificar el marcador a usar para la evaluación y validación y de no existir crear uno propio. Los marcadores proporcionan una vía objetiva de evaluación o de comparación de la solución.

- **Uso de métricas**

El uso de métricas se propone para evaluar el desempeño de la solución y para probar o argumentar las hipótesis que se han hecho en relación con el rendimiento de la misma. Es necesario para su correcta aplicación determinar si existen o no las métricas que son apropiadas para medir su rendimiento y compararlos resultados con las soluciones anteriores (si es que existen). Si tales parámetros no existen, entonces es necesario determinar si existen o no las métricas para medir el desempeño de problemas similares al problema a evaluar. En tal caso, se necesita argumentar que el uso de las métricas elegidas es una forma razonable de evaluación y validación de una solución.

- **Pruebas matemáticas**

Las pruebas matemáticas consisten en demostrar matemáticamente las afirmaciones que se hacen acerca de la solución que se ha desarrollado. Las afirmaciones hipotéticas para esta deben expresarse cuantitativamente, y los aspectos esenciales del problema y su respuesta se pueden expresar formalmente en un sistema lógico cerrado. Este modelo ofrece la forma más fuerte de validación de las afirmaciones hechas sobre la solución.

- **Razonamiento lógico**

El razonamiento lógico es generalmente usado para argumentar la validez de la solución desarrollada y no es posible utilizar una prueba matemática formal para establecer la validez de la misma. El razonamiento lógico es válido también en contextos donde el problema puede ser demasiado complejo, o puede que no sea posible formular el problema y los criterios de respuesta en un marco formal. Las construcciones y los supuestos del problema son, sin embargo, lo suficientemente precisos para pueda construirse un argumento lógico basado en las hipótesis del resultado. Este patrón podría servir como un suplemento o como alternativa a la evaluación experimental. El razonamiento lógico está compuesto por 3 fases fundamentales: la identificación de los supuestos (axiomas), identificación de las reglas (deducciones) y la construcción del modelo lógico.

Se propone para el desarrollo de las pruebas en esta investigación utilizar el método de Demostración porque este puede ser aplicado a soluciones novedosas y que no cuenten con soluciones similares que permitan aplicar otros métodos comparativos.

3.2.1 Características de la muestra.

Para la demostración de la solución se propone un diseño pre-experimental siguiendo la estrategia de pre-prueba/post-prueba con un solo grupo [Sampieri et al. 2006].

Se analizará el comportamiento de la variable calidad de los datos de acuerdo con el indicador de efectividad, estableciéndose comparaciones de acuerdo al escenario anterior y posterior a la aplicación del proceso de limpieza de datos. El grupo de control lo constituyen tres backups reales de la BD del GESPRO, los cuáles van a tener diferentes volúmenes de datos, integrados por tablas de varios campos, asociados a las tareas asignadas y propuestas por los estudiantes y profesores de la universidad. Para cada uno se midió el tiempo de ejecución evaluándose además la reducción de las anomalías existentes y su manera de repercutir en el tamaño de la instancia de entrada al algoritmo de limpieza de datos.

3.2.2 Recursos utilizados para la demostración.

El hardware disponible es una computadora ACPI Multiprocessor PC, con una motherboard P5G41M-LE, procesador Intel Celeron 2.6 a 2.60GHz y una memoria DDR2 con 1GB de capacidad. El sistema operativo sobre el cual se probaron los algoritmos implementados fue Ubuntu v11.4.

3.2.3 Validación de los indicadores en los casos de estudios.

De acuerdo con la muestra seleccionada, se realizó la validación en vista a los indicadores propuestos, logrando así efectuar comparaciones de los resultados propuestos y los resultados logrados realmente, para llegar a conclusiones referentes a la eficiencia de los algoritmos implementados. La comparación de la cantidad de errores detectados antes del proceso de limpieza, y la cantidad erradicada después del proceso se muestran en las figuras 16 y 17.

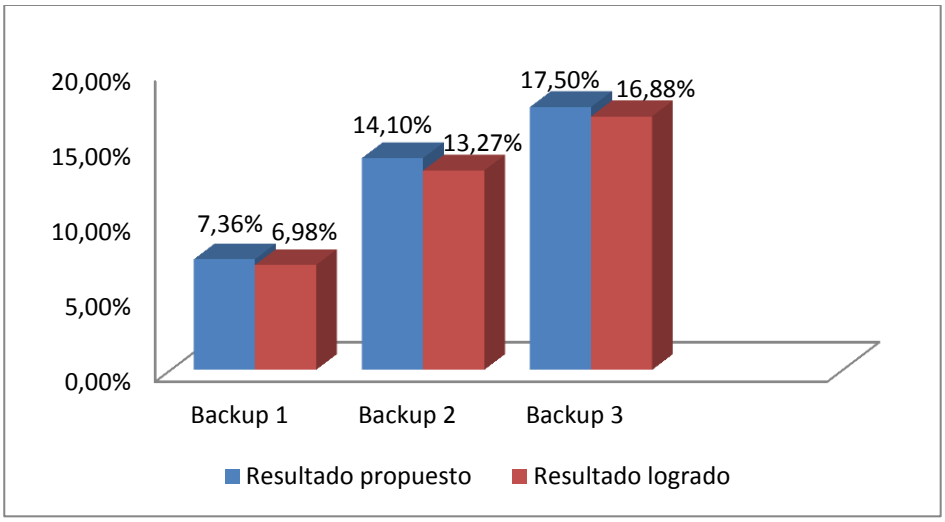


Figura 16: Errores detectados y corregidos con la función de similitud de Jaro

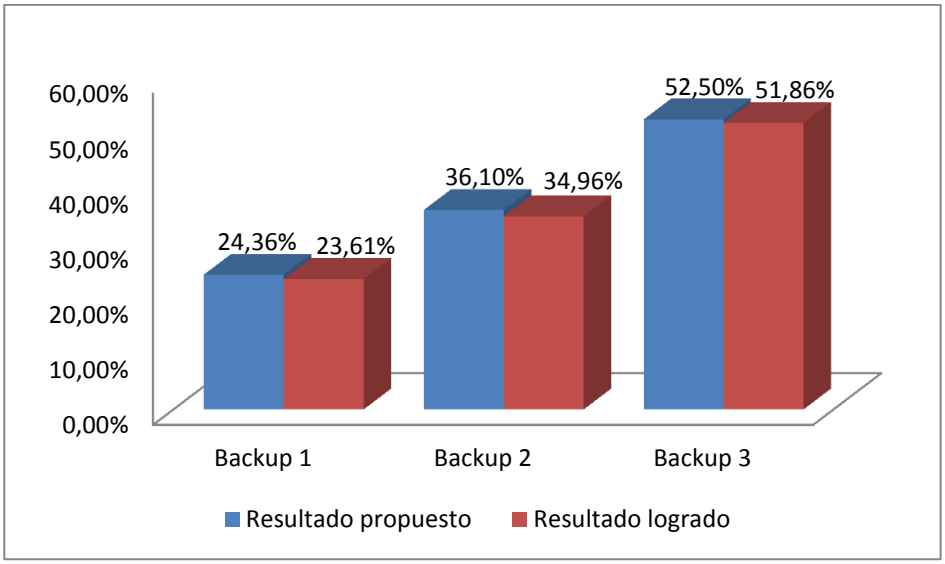


Figura 17: Errores detectados y corregidos con la función de distancia de edición

A continuación se muestra el comportamiento de los indicadores de eficiencia para cada algoritmo implementado. Ver Tabla 5 y 6.

Tabla 5: Descripción de los indicadores según resultados de la similitud de Jaro

Muestra	**Tiempo**	**Volumen**	**Efectividad**
Backup_1	6,79155 min	800 tareas	94.83%

Backup_2	54,52 min	2200 tareas	94.11%
Backup_3	3,06 h	5300 tareas	96.42%

Tabla 6: Descripción de los indicadores según resultados de la distancia de edición

Muestra	Tiempo	Volumen	Efectividad
Backup_1	2.45 min	800 tareas	96.92%
Backup_2	5.46 min	2200 tareas	96.84%
Backup_3	9.57 min	5300 tareas	98.78%

3.2.4 Validación de la calidad de los datos.

Dado a que la efectividad viene expresada como variable de calidad de datos, o sea, la cantidad de errores que pueden ser corregidos por el algoritmo de limpieza de datos implementado, se puede afirmar que a mayor efectividad, será menor la existencia de datos sucios en la BD.

Como se explicó en la sección 3.2.1 se cuenta con tres backups de volúmenes diferentes para realizar la evaluación y validación delos algoritmos de limpieza de datos desarrollados en esta investigación. Por cada conjunto de datos se desarrolló un caso de prueba como se muestra a continuación:

Aplicando el algoritmo de similitud de Jaro para detección de duplicados.

Caso 1: En este caso se seleccionó el backup 1 (ver tabla 5) que cuenta con 16columnas, con un total de800 tareas. Para este conjunto de datos con los recursos descritos en la sección 3.2.2 el tiempo de ejecución del algoritmo fue de 6,79155 minutos, con un porciento de efectividad de 94.83%, eliminando 123 duplicados.

Caso 2: En este caso se seleccionó el backup2 (ver tabla 5) que cuenta con 16 columnas, con un total de 2200 tareas. Para este conjunto de datos con los recursos descritos en la sección 3.2.2 el tiempo de ejecución del algoritmo fue de 54.52 minutos, con un porciento de efectividad de 94.11%, eliminando 270 duplicados.

Caso 3: En este caso se seleccionó el backup3 (ver tabla 5) que cuenta con 16 columnas, con un total de 5300 tareas. Para este conjunto de datos con los recursos descritos en la sección 3.2.2 el tiempo de ejecución del algoritmo fue de 3.06 horas, con un porciento de efectividad de 96.42%, eliminando 341 duplicados.

Aplicando el algoritmo de distancia de edición para corrección de errores ortográficos, no estandarización de cadenas e irregularidades.

Caso 1: En este caso se seleccionó el backup 1 (ver tabla 6) que cuenta con 16 columnas, con un total de 800tareas. Para este conjunto de datos con los recursos descritos en la sección 3.2.2 el tiempo de ejecución del algoritmo fue de 2.45 minutos, con un porciento de efectividad de 96.92%.

Caso 2: En este caso se seleccionó el backup 2 (ver tabla 6) que cuenta con 16 columnas, con un total de 2200 tareas. Para este conjunto de datos con los recursos descritos en la sección 3.2.2 el tiempo de ejecución del algoritmo fue de 5.46 minutos, con un porciento de efectividad de 96.84%.

Caso 3: En este caso se seleccionó el backup 3 (ver tabla 6) que cuenta con 16 columnas, con un total de 5300 tareas. Para este conjunto de datos con los recursos descritos en la sección 3.2.2 el tiempo de ejecución del algoritmo fue de 9.57 minutos, con un porciento de efectividad de 98.78%

Se puede comprobar que la variación de la calidad del dato mejoró en cada una de las muestras probadas y que el algoritmo desarrollado para la solución propuesta funciona correctamente.

3.3 Realización de un informe comparativo del estado de los datos antes y después de la limpieza.

Se puede constatar por los resultados obtenidos en las pruebas realizadas como se logran erradicar un gran porciento de los errores detectados en la BD, por tanto se puede afirmar que la efectividad de los algoritmos propuestos es alta para la solución a los problemas detectados. Se puede notar cómo, a mayor volumen de datos de entrada, mayor será el tiempo de ejecución del algoritmo de limpieza de datos aplicado. Ver figuras 18 y 19 para observar el nivel de errores antes y después de la limpieza y ver la figura 20 para observar el grado de efectividad de los algoritmos implementados.

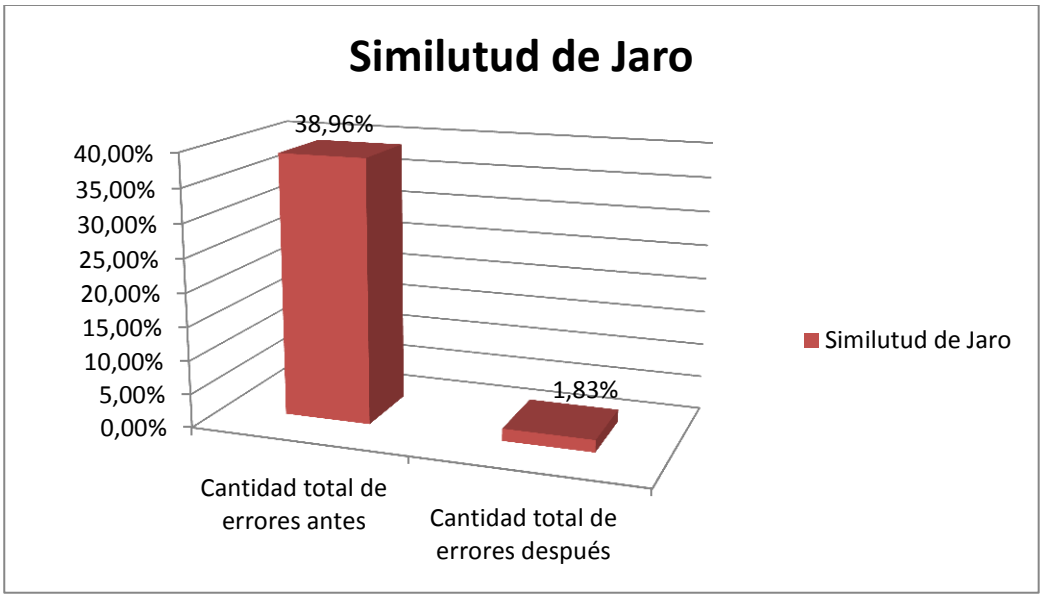


Figura 18: Comparación de porciento de errores antes y después de la limpieza de datos

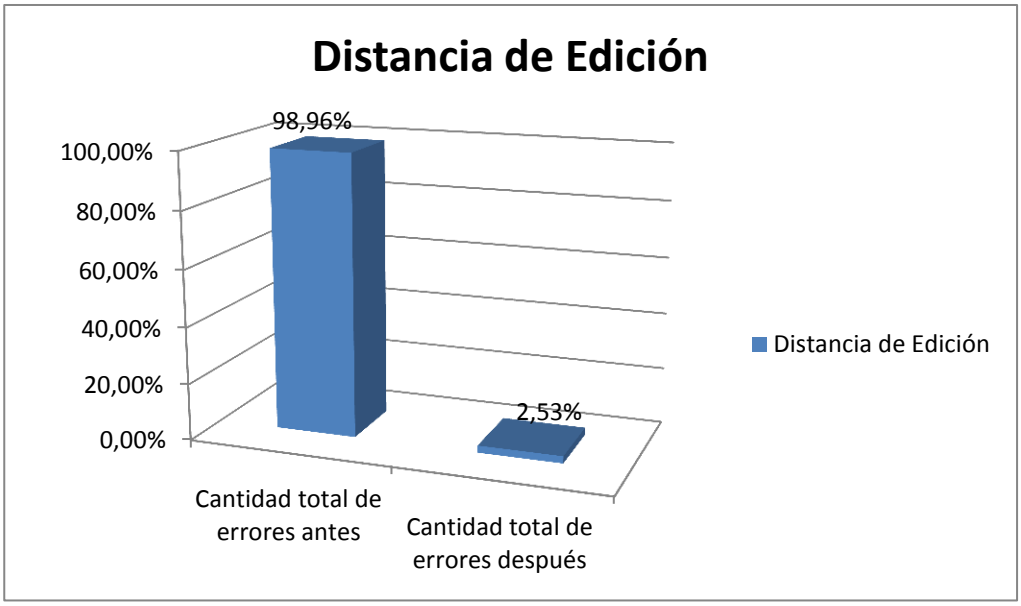


Figura 19: Comparación de porciento de errores antes y después de la limpieza de datos

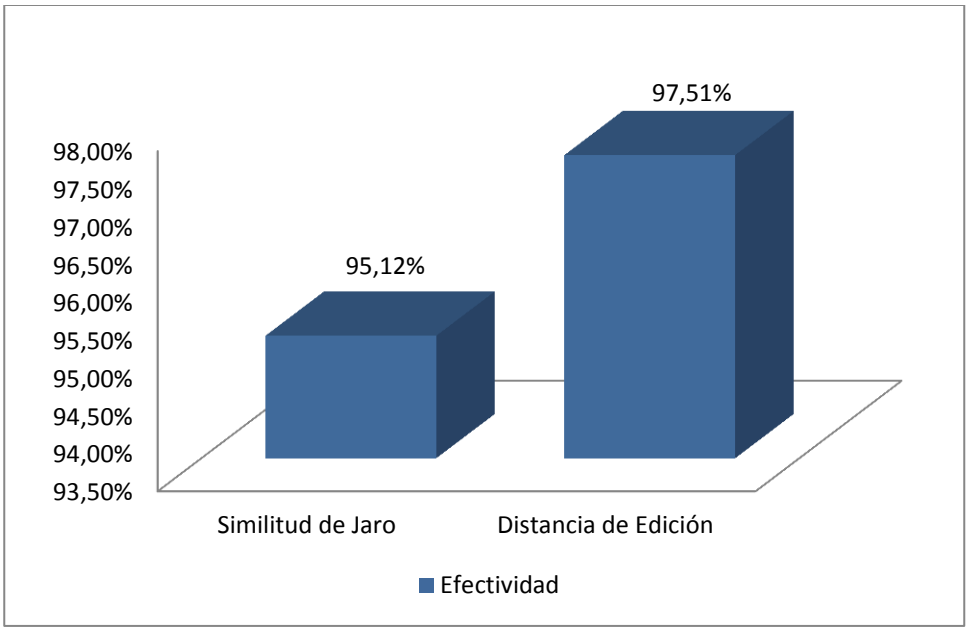


Figura 20: Efectividad

Conclusiones Parciales

- A partir de las pruebas y estudios realizados a los algoritmos implementados, se puede constatar que los mismos ayudan a garantizar la calidad de los datos que almacenan, ya que su aplicación arrojó resultados satisfactorios, con un promedio del 96% de corrección de errores.

- Se realizó la validación de los algoritmos en el entorno de aplicación de los mismos, proponiendo indicadores para medir la eficiencia en cuanto a tiempo de ejecución, según el volumen de datos y la efectividad de estos a la hora de corregir las anomalías detectadas en la BD de datos del GESPRO, obteniéndose de los mismos que para su ejecución en grandes cantidades de información estos tienden a tardar un poco, recomendándose su ejecución en horas de poco uso de la BD.

CONCLUSIONES

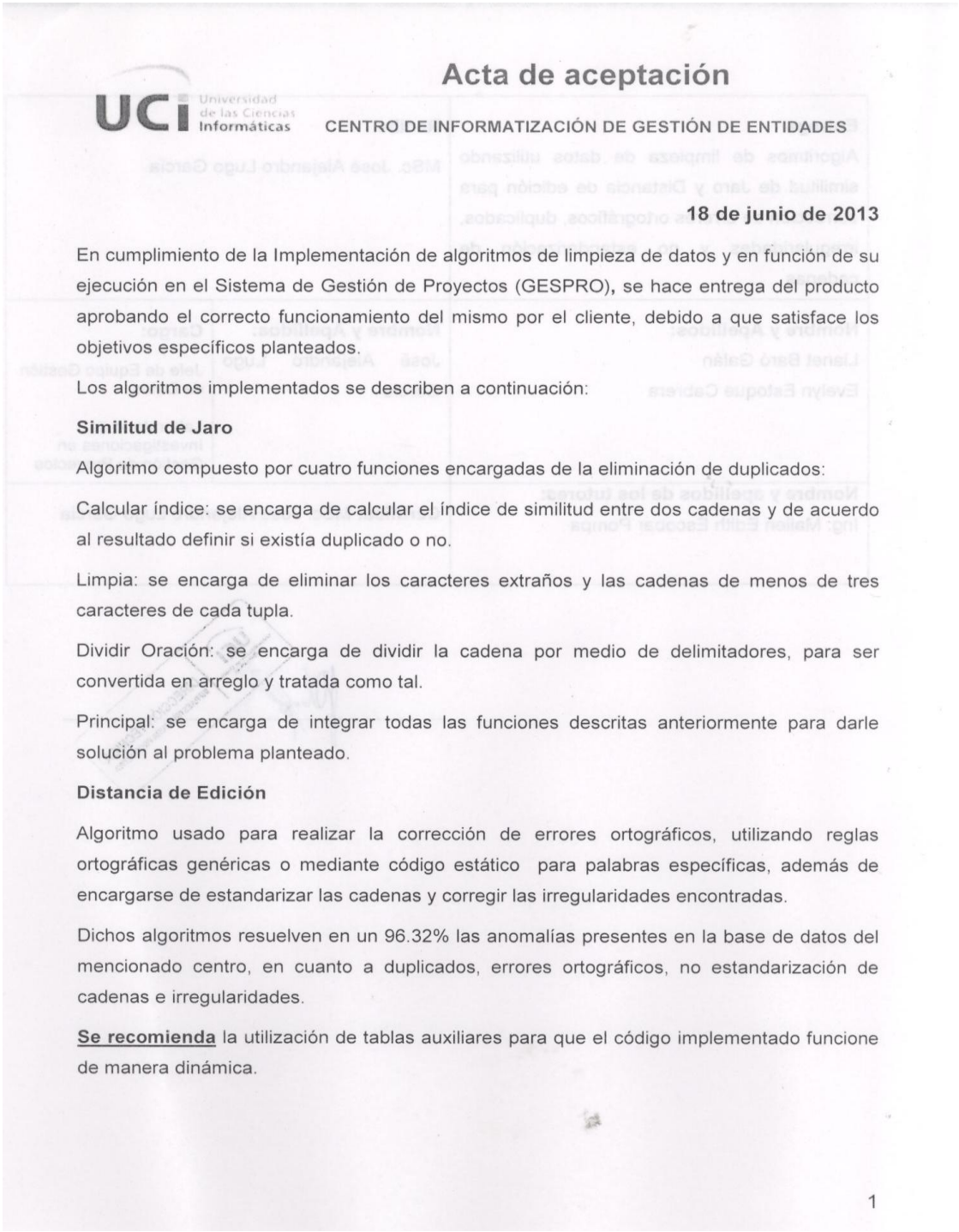
- El estudio del estado del arte permitió sentar las bases teóricas para la investigación, obteniéndose una taxonomía de errores comunes y escogiéndose los algoritmos para erradicar anomalías como irregularidades, errores ortográficos, duplicados y no estandarización de cadenas.
- La metodología y tecnologías utilizadas permitieron guiar e implementar el proceso de limpieza de datos.
- Con la implementación de los algoritmos de similitud de Jaro y distancia de edición se logró concretar la limpieza de datos a realizar en la BD del GESPRO.
- La evaluación de los indicadores de eficiencia identificados para los algoritmos permitió validar la calidad de los mismos para el entorno del GESPRO, concluyéndose que el objetivo general de la investigación fue cumplido satisfactoriamente.

RECOMENDACIONES

- Se recomienda que se utilicen las funciones implementadas en momentos que haya poco tráfico de usuarios en la herramienta; por las noches, feriados o fines de semana.
- Se propone integrar los algoritmos en una herramienta para la creación de una futura suite para realizar la limpieza de datos.

ANEXOS

Anexo 1: Acta de aceptación del producto ´´Algoritmos de limpieza de datos´´



UCI Universidad de las Ciencias Informáticas

Acta de aceptación

CENTRO DE INFORMATIZACIÓN DE GESTIÓN DE ENTIDADES

18 de junio de 2013

En cumplimiento de la Implementación de algoritmos de limpieza de datos y en función de su ejecución en el Sistema de Gestión de Proyectos (GESPRO), se hace entrega del producto aprobando el correcto funcionamiento del mismo por el cliente, debido a que satisface los objetivos específicos planteados.

Los algoritmos implementados se describen a continuación:

Similitud de Jaro

Algoritmo compuesto por cuatro funciones encargadas de la eliminación de duplicados:

Calcular índice: se encarga de calcular el índice de similitud entre dos cadenas y de acuerdo al resultado definir si existía duplicado o no.

Limpia: se encarga de eliminar los caracteres extraños y las cadenas de menos de tres caracteres de cada tupla.

Dividir Oración: se encarga de dividir la cadena por medio de delimitadores, para ser convertida en arreglo y tratada como tal.

Principal: se encarga de integrar todas las funciones descritas anteriormente para darle solución al problema planteado.

Distancia de Edición

Algoritmo usado para realizar la corrección de errores ortográficos, utilizando reglas ortográficas genéricas o mediante código estático para palabras específicas, además de encargarse de estandarizar las cadenas y corregir las irregularidades encontradas.

Dichos algoritmos resuelven en un 96.32% las anomalías presentes en la base de datos del mencionado centro, en cuanto a duplicados, errores ortográficos, no estandarización de cadenas e irregularidades.

Se recomienda la utilización de tablas auxiliares para que el código implementado funcione de manera dinámica.

1

<table>
<tr><td>Entrega
Algoritmos de limpieza de datos utilizando similitud de Jaro y Distancia de edición para corrección de errores ortográficos, duplicados, irregularidades y no estandarización de cadenas.</td><td colspan="2">Recibe
MSc. José Alejandro Lugo García</td></tr>
<tr><td>Nombre y Apellidos:
Lianet Baró Galán
Evelyn Estoque Cabrera</td><td>Nombre y Apellidos:
José Alejandro Lugo García</td><td>Cargo:
Jefe de Equipo Gestión de Datos.

Laboratorio de Investigaciones en Gestión de Proyectos</td></tr>
<tr><td>Nombre y apellidos de los tutores:
Ing. Mailen Edith Escobar Pompa</td><td colspan="2">Certifica: MSc. José Alejandro Lugo García</td></tr>
</table>

REFENCIAS BIBLIOGRÁFICAS

ABAD, M.T. Análisis de la eficiencia de los algoritmos. In.: Departament de Llenguatges i Sistemes Informàtics, FIB-UPC, 2011, p. 24.

ABITEBOUL, S., CLUET, S., MILO, T., MOGILEVSKY, P., SIMEON, J. AND ZOHAR, S. Tools for Data Translation and Integration. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 1999.

AHO, A.V. AND ULLMAN, J.D. *Principles of Compiler Design*. Edtion ed.: Addison-Wesley 1979. 604 p.

AMÓN, I. Guía Metodológica para la selección de técnicas de depuración de datos. In *Facultad de Minas, Escuela de Sistemas Medellin.* Medellín, Colombia: Universidad Nacional de Colombia, 2010, p. 120.

CABRERA LAMADRID, G.G. AND CADENAS DEL LLANO, P.F. Herramienta para evaluar factibilidad en proyectos de software. In *Facultad 3.* La Habana: Universidad de las Ciencias Informáticas, 2012, p. 83.

CÁCERES, A.A. La métrica de Levenshtein. Revista de Ciencias Básicas, Universidad Juárez Autónoma de Tabasco, 2008, vol. 7, no. 2, p. 35-43.

CHRISTEN, P. A Comparison of Personal Name Matching: Techniques and Practical Issues. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops.* IEEE Computer Society, 2006, p. 5.

DASU, T., VESONDER, G.T. AND WRIGHT, J.R. 2003. Data quality through knowledge engineering. In *Proceedings of the Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, D.C.2003 ACM, 956844, 705-710.

EĞECIOĞLU, O. AND IBEL, M. Parallel Algorithms for Fast Computation of Normalized Edit Distances. In *Proceedings of the 8th IEEE Symposium on Parallel and Distributed Processing.*1996, p. 496-503.

ESCOBAR, M.E., PIÑERO, Y. AND PACHECO, E. Base de casos para la evaluación de competencias de equipos a partir de evidencias. In *UCIENCIA 2012.* UCI, 2012.

GALHARDAS, H., FLORESCU, D., SHASHA, D. AND SIMON, E. AJAX: an extensible data cleaning tool. ACM, 2000.

GALHARDAS, H., FLORESCU, D., SHASHA, D., SIMON, E. AND SAITA, C.-A. 2001. Declarative Data Cleaning: Language, Model, and Algorithms. In *Proceedings of the Proceedings of the 27th International Conference on Very Large Data Bases*2001 Morgan Kaufmann Publishers Inc., 672042, 371-380.

GOTOH, O. An Improved Algorithm for Matching Biological Sequences. Journal of Molecular Biology, 1982, vol. 162, no. 3, p. 705-708.

HAN, J. AND KAMBER, M. *Data Mining: Concepts and Techniques. (The Morgan Kaufmann Series in Data Management Systems).* Edtion ed.: Morgan Kaufmann, 2000.

HERNÁNDEZ, M.A. The merge/purge problem for large databases. In *Proceedings of the ACM SIGMOD Conference.*1995.

HERNÁNDEZ, Y.V. DBAnalyzer 2.0, sistema para analizar bases de datos libres. In *Facultad de Software Libre.* La Habana: Universidad de las Ciencias Informáticas., 2008, p. 49.

HYYRÖ, H. A Bit-Vector Algorithm for Computing Levenshtein and
Damerau Edit Distances. In *The Prague Stringology Conference '02.* 2012.

JARO, M.A. Unimatch: A Record Linkage System: User's Manual, technical report. In. Washington, D.C.: US Bureau of the Census, 1976.

KIM, W., CHOI, B.-J., HONG, E.-K., KIM, S.-K. AND LEE, D. A Taxonomy of Dirty Data. Data mining and knowledge discovery, 2003, vol. 7, no. 1, p. 81-99.

LEE, M.L., LING, T.W. AND LOW, W.L. Intelliclean: A knowledge-based intelligent data cleaner. In *Proceedings of the sixth ACM SIGKDD international conference of Knowlegde discovery and data mining.*2000, p. 290-294.

LÓPEZ, B. "Limpieza de Datos: Reemplazo de valores ausentes y Estandarización". Resumen de la tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. In *Facultad de Matemática y Computación.* Santa Clara: Universidad Central "Marta Abreu" de Las Villas, 2011, p. 45.

LOWRENCE, R. AND WAGNER, R.A. An Extension of the String-to-String Correction Problem. Journal of the ACM, 1975, vol. 22, no. 2, p. 177-183.

MALETIC, J.I. AND MARCUS, A.*Cap. 2 A Prelude to Knowledge Discovery. The Data Mining and Knowledge Discovery Handbook.* Edtion ed.: O. Maimon, L. Rokach: Springer, 2005.

MARMOL, D. AND LÓPEZ, B. Determinación de una taxonomía de errores en los sistemas operacionales de nuestro entorno. In. Santa Clara: Departamento Ciencia de la Computación, Universidad Central de las Villas, 2005.

MÁRQUEZ, M.A. Programación .NET y PostgreSQL en la consola. Tutoriales y código acerca de la plataforma .NET,Redes con Cisco y PostgreSQL. 2012, [cited 3 de mayo 2013]. Available from Internet:<http://xomalli.blogspot.com/2010/04/uso-de-funciones-plsql-en-postgresql.html>.

MARTÍNEZ VIGIL, I. Módulo de Gestión de Riesgos versión 2.0 para el sistema GESPRO 12.05. In *Laboratorio de Gestión de Proyectos, Facultad 5*. La Habana: Universidad de las Ciencias Informáticas, 2012, p. 80.

MARZAL, A. AND VIDAL, E. Computation of Normalized Edit Distance and Applications. IEEE Transactions on Patern Analysis and Machine Intelligence, 1993, vol. 5, no. 9.

MASEK, W.J. A Faster Algorithm for Computing String Edit Distances. Journal of Computer and System Sciences, 1980, vol. 20, p. 18-31.

MAYOL, E. AND TENIENTE, E. 1999. A Survey of Current Methods for Integrity Constraint Maintenance and View Updating. In *Proceedings of the Proceedings of the Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling*1999 Springer-Verlag, 728206, 62-73.

MÜLLER, H. AND FREYTAG, J.-C. Problems, Methods, and Challenges in Comprehensive Data Cleansing. In *Technical Report HUB-IB-164*,. Berlin: Humboldt University Berlin,, 2003, p. 23.

NEEDLEMAN, S.B. AND WUNSH, C.D. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins 1970, vol. 48, no. 3.

NORMALIZACIÓN, O.N.D. Sistema de Gestión Integrada de Capital Humano - Vocabulario. In. La Habana, Cuba, 2007, vol. NC 3000:2007.

PANIAGUA, J., MIRA , J.F. AND AMÓN, I. Elaboración de diagnostico de calidad de datos para una empresa del sector salud. *Universidad Pontificia Bolivariana* [Type of Work]. 2010. Available from Internet:<http://kosmos.upb.edu.co/web/uploads/articulos/%28A%29_Diagnostico_de_la_calidad_de_la_base_de_datos_De_la_Clinica_Universitaria_Bolivariana_NySsg.pdf>.

PÉREZ QUINTERO, L. "Modelo para la evaluación por competencias en proyectos informáticos de la Universidad de las Ciencias Informáticas". In. Cuidad de la habana: Universidad de las Ciencias Informáticas, 2010, p. 132.

PGADMIN. pgAdmin PostgreSQL Tools. Introduction. In.: http://www.pgadmin.org/, 2010.

PIÑERO PÉREZ, P.Y., PESTANO PINO, H., VÁZQUEZ ACOSTA, M., ABELARDO, F.N., LUGO, J.A., MÉNDEZ ROLDÁN, I., AHMED, E., MENÉNDEZ RIZO, J., PIÑERO, P.R., IZQUIERDO, M., TORRES, S., PÉREZ, A.D. AND GONZÁLEZ JORRÍN, M. Experiencias en el uso de PostgreSQL en el sistema GESPRO, un enfoque práctico. Revista Cubana de Ciencias Informáticas (RCCI), Septiembre 2011, p. 10.

POSTGRESQL. PL/pgSQL - SQL Procedural Language. 2003, [cited Marzo 2013]. Available from Internet:<ftp://www.cc.uah.es/pub/Alumnos/I.Informatica/Fund.Bases.Datos/Laboratorio/Sesion5/pl-pgsql.pdf>.

POSTGRESQL. PostgreSQL 9.1.9 Documentation. In.: Sitio oficial del Grupo Global de Desarrollo de PostgreSQL. http://www.postgresql.org, 2013.

RAE. Diccionario de la Real Academia Española. In. España., 2012

RAHM, E. AND DO, H.H. Data Cleaning: Problems and Current Approaches. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2000, p. 11.

RAMAN, V. AND HELLERSTEIN, J.M. Potter'sWheel: An Interactive Data Cleaning System. In *Proceedings of the 27th VLDB Conference.* Roma, Italia, 2001, p. 10.

REDMAN, T.C.*Data Quality. The Field Guide*. Edtion ed. Boston, Estados Unidos: Digital Press, 2001. ISBN 1-55558-251-6.

RISTAD, E. AND YIANILOS, P. Learning string edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, vol. 20, no. 5, p. 522-532.

ROCHNIK, N. AND DIJCKS, J.-P. Oracle Warehouse Builder 10gR2 Transforming Data into Quality Information. 2006, [cited 2013 6 de abril], pp. 16. Available from Internet:<http://www.oracle.com/technetwork/developer-tools/warehouse/transforming-1.pdf>.

SAMPIERI, R.H., COLLADO, C.R. AND LUCIO, P.B.*Metodología de la investigación*. Edtion ed. Mexico, D.F., Mexico: McGraw-Hill Interamericana, 2006. 882 p. ISBN 970-10-5753-8.

SÁNCHEZ GIRAUT, R. Ingeniería de Requisitos de los módulos Gestión Logística, Gestión Documental, Trabajo Colaborativo y Configuración del sistema para la Gestión de Proyectos GESPRO 12.05. In *Laboratorio de Gestión de Proyectos (GESPRO), Facultad 5.* La Habana: Universidad de las Ciencias Informáticas, 2012, p. 78.

SATTLER, K.-U., CONRAD, S. AND SAAKE, G. Adding conflict resolution features to a query language for database federations. In *Proc. 3nd Int. Workshop on Engineering Federated Information Systems, EFIS'00.* Dublin, Ireland, 2000, p. 41-52.

SATTLER, K.-U. AND SCHALLEHN, E. A data preparation framework based on a multidatabase language. In *Database Engineering & Applications, International Symposium.* 2001, p. 219 - 228.

SMITH, T.F. AND WATERMAN, M.S. Identification of Common Molecular Subsequences. Journal of Molecular Biology, 1981, vol. 147, no. 1, p. 195-197.

TIERSTEIN, L.M. A Methodology for Data Cleansing and Conversion. 2005, [cited 2013 6 de abril], pp. 21. Available from Internet:<http://www.google.com.cu/url?sa=t&rct=j&q=TIERSTEIN%2C+Leslie+M.+A+Methodology+for+Data+Cleansing+and+Conversion.+2005.&source=web&cd=1&ved=0CC8QFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.113.2362%26rep%3Drep1%26type%3Dpdf&ei=quxiUYvkIPLI4APgwYDICQ&usg=AFQjCNFUQWZG-l4Q_XBMnM0syF15Ap_nkQ&cad=rja>.

UC3M. Inteligencia Artificial para mejorar el procesamiento de datos. 2011, [cited 22 de octubre 2012]. Available from Internet:<http://www.agenciasinc.es/Noticias/Inteligencia-Artificial-para-mejorar-el-procesamiento-de-datos.>.

VAISHNAVI, V.K. AND WILLIAM KUECHLER, J.*Design Science Research Methods and Patterns: Innovating Information and Communication Technology*. Edtion ed.: Auerbach Publications, 2007. 248 p. ISBN 1420059327, 9781420059328.

VASSILIADIS, P., VAGENA, Z., SKIADOPOULOS, S., KARAYANNIDIS, N. AND SELLIS, T. Arktos: towards the modeling, design, control and execution of ETL processes. Information Systems, Elsevier, 2001, vol. 26, no. 8, p. 25.

WAGNER, R.A. AND FISCHER, M.J. The String-to-String Correction Problem. Journal of the ACM, 1974, vol. 21, no. 1, p. 168-173.

WANG, R.Y. AND STRONG, D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. Journal of Management Information Systems, Spring, 1996, vol. 12, no. 4, p. 5-33.

YANCEY, W.E. Evaluating String Comparator Performance for Record Linkage. In *Proceedings of the Fifth Australasian Conference on Data mining and Analytics.* Australia, 2006, p. 23-21.

YUJIAN, L. AND BO, L. A Normalized Levenshtein Distance Metric. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, vol. 29, no. 6, p. 1091-1095.