UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS
FACULTAD DE MATEMÁTICA FÍSICA Y COMPUTACIÓN
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

LIMPIEZA DE DATOS:
REEMPLAZO DE VALORES AUSENTES Y ESTANDARIZACIÓN

Tesis presentada en opción al grado científico de
Doctor en Ciencias Técnicas

BEATRIZ LÓPEZ PORRERO

Santa Clara

2011

UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS
FACULTAD DE MATEMÁTICA FÍSICA Y COMPUTACIÓN
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

LIMPIEZA DE DATOS:
REEMPLAZO DE VALORES AUSENTES Y ESTANDARIZACIÓN

Tesis presentada en opción al grado científico de
Doctor en Ciencias Técnicas

Autora: MSc. Beatriz López Porrero
Tutor: Dr.C. Ramiro A. Pérez Vázquez

Santa Clara

2011

AGRADECIMIENTOS

A mi tutor y esposo Dr.C. Ramiro Pérez Vázquez por su paciencia y amor.

A mis hijas Betty y Gabi por impulsarme a cumplir con este proyecto de alcanzar mi doctorado.

A mis padres y mi hermano por estar ahí.

A mis compañeros del Departamento, ¡todos!.

A nuestro Comandante en Jefe Fidel Castro, a la Revolución y a mi familia!!!

SINTESIS

La limpieza de datos es un proceso de gran importancia cuando se quiere asegurar la calidad de los mismos. Aunque existen varias herramientas que permiten realizar distintas tareas relacionadas con la limpieza de datos, por diferentes circunstancias estas no son de fácil acceso para los especialistas de nuestro país.

En este trabajo se hace un estudio de los principales tipos de errores que pueden aparecer en las bases de datos, proponiéndose una taxonomía de errores para las bases de datos cubanas, en que se constata que algunos de los que más se presentan son la forma no estándar de representar campos de tipo cadenas de caracteres y la ausencia de información. Se propuso una modificación de la distancia de edición de Levenshtein, un marco de trabajo para la estandarización de cadenas de caracteres y un método de trabajo para realizar en el ambiente de las bases de datos el reemplazo de los valores ausentes. Se obtuvieron las herramientas DBAnalyzer, que ayuda a la detección de errores en los datos, DBStandard, que permite la estandarización de las cadenas de caracteres, aplicando una modificación a la distancia de edición de Levenshtein y DbNulos, que ofrece un asistente que ayuda al especialista a la selección de los métodos para la sustitución de los valores ausentes.

Las herramientas han sido aplicadas en varias empresas que validan la efectividad del uso de las mismas, como por ejemplo en las bases de datos de Recursos Humanos de la Universidad Central, de la ONAT (Oficina Nacional de Administración Tributaria) de Ranchuelo, del Departamento de Anatomía patológica del Hospital Provincial de Villa Clara y otras.

INTRODUCCIÓN 1

1. LA LIMPIEZA DE DATOS 10

1.1. DEFINICIÓN DE ERROR O ANOMALÍA EN LOS DATOS. 11

1.1.1. Clasificaciones de los errores o anomalías en los datos. 11

1.1.2. La anomalía de valor ausente. 13

1.1.3. La anomalía de falta de estándares en la representación de los datos. 15

1.1.3.1. La estandarización de datos tipo cadenas 16

1.2. EL PROCESO DE LIMPIEZA DE DATOS 20

1.2.1. La limpieza de datos en los almacenes de datos 21

1.2.2. La limpieza de datos en el área de Calidad de los Datos 25

1.2.3. La limpieza de datos en el proceso de KDD 25

1.3. ANÁLISIS DE DATOS 27

1.3.1. El perfil de los datos 28

1.3.2. Minería de datos 28

1.4. MÉTODOS GENERALES USADOS EN LA LIMPIEZA DE DATOS 29

1.4.1. Métodos empleados en el tratamiento de los valores ausentes 30

1.4.2. Métodos para la estandarización de cadenas de textos 31

1.5. HERRAMIENTAS PARA LA REALIZACIÓN DE LA LIMPIEZA DE DATOS. 34

1.6. CONCLUSIONES PARCIALES 36

2. DESCRIPCIÓN DE LAS SOLUCIONES PROPUESTAS PARA LA LIMPIEZA DE DATOS 38

2.1. TAXONOMÍA DE ERRORES 38

2.2. SOLUCIÓN PROPUESTA PARA LA ESTANDARIZACIÓN DE LOS CAMPOS TIPO CADENA DE CARACTERES. 45

2.2.1. Nueva distancia de edición *49*

2.2.2. Estandarización de las direcciones postales. *57*

2.3. SOLUCIÓN PROPUESTA PARA EL TRATAMIENTO DE LOS VALORES AUSENTES 61

2.4. CONCLUSIONES PARCIALES 68

3. APLICACIONES Y EXPERIMENTACIÓN 70

3.1. CONCEPCIÓN DE LAS HERRAMIENTAS Y SUS CARACTERÍSTICAS GENERALES 70

3.1.1. El DBAnalyzer *70*

3.1.2. El DBStandard *78*

3.1.2.1. Experimentos realizados *79*

3.1.2.2. Estandarización de Direcciones *88*

3.1.3. El DBNulos *89*

3.1.3.1. Concepción de la herramienta DBNulos *89*

3.1.3.2. Experimentación *90*

3.2. CONCLUSIONES PARCIALES 97

CONCLUSIONES 98

RECOMENDACIONES 99

REFERENCIAS BIBLIOGRÁFICAS 100

ANEXO A 118

ANEXO B 121

ANEXO C 122

ANEXO D 127

ANEXO E 132

ANEXO F 134

ANEXO G 144

INTRODUCCIÓN

Los sistemas de información en la era actual abarcan importantes esferas como son la economía, el gobierno y las investigaciones científicas. Estas aplicaciones se caracterizan por el manejo de enormes cantidades de datos.

La existencia de anomalías e impurezas en estas grandes cantidades de datos aparece como un fenómeno que distorsiona los resultados que se obtienen de la interpretación y el análisis de ellos, y provocan como consecuencias, la elevación de los costos y la disminución de los beneficios de su procesamiento.

En los sistemas que manejan grandes volúmenes de datos la calidad de los mismos ha constituido una preocupación y ocupación de la comunidad científica. Se ha podido establecer que, de entre los disímiles aspectos que influyen en la calidad de los datos, la fuente desde donde provienen constituye un factor crucial. Los procesos de adquisición y entrada de datos, independientemente de su complejidad, son muy propensos a aumentar la cantidad de errores. Los desarrolladores de sistemas dedican ingentes esfuerzos a disminuirlos, pero el hecho es que los índices de errores se mantienen elevados. Redman en [118] planteó que, aún habiéndose usado los métodos más sofisticados para tratar de evitarlos, la razón de error en los datos es de alrededor de un 5%. En [31] se señala que en las bases de datos alrededor del 60% de los datos tienen algún problema de calidad. Un estudio más reciente señala que el 25% de los datos críticos de una empresa presentan errores y que las empresas en muchas ocasiones no tienen conciencia de ello [47].

La realidad cubana no escapa a esta situación, cada vez se hace más necesario que las empresas automaticen sus procesos o el control de estos, lo que conduce a la creación de

sistemas de información en los que, en ocasiones, no se garantiza la calidad de los datos que se almacenan y procesan; aparecen, por ejemplo, listas de clientes y direcciones en que el nombre de una misma calle o entidad se expresa de formas distintas, o aparecen otras inconsistencias o sencillamente falta información.

La limpieza de datos, como se conoce al proceso que se encarga de corregir los errores en los datos, se convierte por tanto en un mecanismo necesario para que las estadísticas, los informes y en última instancia las decisiones que se tomen por los directivos sean confiables, pues en la medida en que esté garantizada la calidad de los datos, así mismo habrá seguridad y fiabilidad en las acciones posteriores que se produzcan a partir de su análisis.

Existen diferentes enfoques que son abordados con relación a la limpieza de datos, uno de estos aspectos, que resulta muy interesante, es la búsqueda en el contexto de lo que se denomina en la literatura "datos sucios" [38, 55, 66]. La búsqueda de contexto es clasificada como el primer paso para entender el proceso de limpieza de datos. En varios trabajos se presentan propuestas de taxonomías de errores en los datos [65, 93, 104], por lo que en el presente trabajo se pretende establecer la concepción de lo que debe ser considerado como datos sucios en los sistemas empresariales cubanos y a partir de allí, resolver los que sean más frecuentemente identificados.

En la literatura se encuentran varias definiciones del término limpieza de datos y casi nunca estas coinciden, pues dependen del área en que es aplicado el proceso. Tres áreas importantes donde este proceso forma parte de su definición son: los almacenes de datos (DW, por las siglas en inglés de DataWarehouse), la calidad de datos (TQMD, Total

Quality Management Data) y el descubrimiento de conocimientos (KDD, Knowledge Data Discovery).

En los almacenes de datos, la limpieza se aplica típicamente para eliminar la aparición de registros duplicados al mezclar los datos que provienen de diferentes orígenes. Esta dirección del proceso de limpieza se conoce como identificación y eliminación de duplicados. Se han desarrollado diferentes métodos para llevarla a cabo [15, 23, 24, 37, 40-42, 74, 131].

En el área de TDQM la limpieza de datos adquiere una relevancia especial para la comunidad científica y de negocios. En este marco, el proceso de limpieza de datos es aquel que trata de "precisar el grado de corrección en los datos y mejorar su calidad" [38].

Por otra parte, en el área de KDD la limpieza de datos se define como el primer paso o preprocesamiento [14, 51, 136]. Varios sistemas de KDD y minería de datos resuelven las actividades de limpieza de datos con herramientas que dependen de dominios específicos. En las últimas décadas se ha producido un auge en el trabajo científico dedicado al estudio de las técnicas, métodos y herramientas para realizar esta importante tarea de limpiar los datos. Además, según [10] las empresas gastan alrededor del 40% de su presupuesto en el aseguramiento de la calidad de sus datos.

Actualmente existen numerosos sistemas y herramientas para resolver la limpieza de datos, pero la gran mayoría están diseñados con dependencia a un dominio específico, necesitan diccionarios ortográficos, geográficos o elementos propios del entorno para su uso, además de que su costo es demasiado elevado y que las propias compañías que los

producen brindan sus servicios. A esto debe añadirse la situación de bloqueo a que está sometido nuestro país que incluye la restricción de adquirirlos.

La falta de estandarización en cadenas de textos y la presencia de valores ausentes son algunas de las manifestaciones de "suciedad" que más se reiteran al manejar grandes volúmenes de datos.

Se entiende en este contexto por falta de estandarización el hecho de que un mismo objeto sea representado o nombrado, en el conjunto de datos, de formas diferentes, sin uniformidad en su representación. La mayoría de las veces esta ausencia de estándar se debe a errores tipográficos al entrar los datos a los sistemas, lo que se produce de manera sistemática, aún cuando los diseñadores de sistemas utilicen las técnicas más sofisticadas para evitarlo. La falta de estándares hace que todas las estadísticas donde los datos se utilicen sean incorrectas o que haya que hacer consultas muy complejas para poder abarcar todas las formas posibles en que ha sido nombrado un objeto. Elmagarmid [35] plantea que la estandarización, es un paso necesario para poder realizar un proceso de mezcla/depuración con calidad.

El uso de distancias o métricas, como por ejemplo la distancia de edición de Levensthein [77], empleadas en algoritmos de agrupamientos, se encuentra entre las soluciones reportadas en la literatura para este fenómeno [19, 27, 28, 35, 102].

El problema de los valores ausentes también es común en grandes volúmenes de datos y puede tener diversas causas. Kimball [67] señala que a un almacén de datos no deben llegar valores ausentes ya que ellos traen múltiples problemas en el momento de hacer estadísticas, buscar tendencias, etc. En la mayoría de las técnicas estadísticas que trabajan sobre grandes volúmenes de datos también hay numerosas técnicas para sustituirlos, en

muchos casos el investigador tiene que decidir qué hacer con ellos. En todos estos casos hay que tener un conocimiento no solo de los datos, sino de los procedimientos estadísticos para trabajar con los datos ausentes. Estas técnicas y procedimientos aparecen en paquetes estadísticos que no son de uso común por los especialistas de las empresas que necesitan reemplazar sus datos ausentes para; por ejemplo, poblar un almacén y en lugar de reemplazarlos los dejan como valores ausentes o eliminan los registros donde estos aparecen.

Con estos elementos estudiados el presente trabajo estará dirigido hacia el cumplimiento del siguiente **Objetivo General**:

Definir procedimientos de limpieza de datos que permitan, en particular, la estandarización de cadenas de caracteres y la sustitución de los valores ausentes, con lo cual se podrá contribuir al mejoramiento de la calidad de los datos en los sistemas de información empresarial.

Como **objetivos específicos** se plantean:

- Determinar una taxonomía de errores en los datos, aplicable a las bases de datos del entorno empresarial cubano
- Ofrecer una solución para la estandarización de datos de tipo cadena donde predominen los errores tipográficos
- Brindar un procedimiento que asista en la sustitución de los valores ausentes
- Elaborar y validar herramientas de limpieza de datos que permita el reemplazo de valores ausentes y la estandarización de la información.

Para la consecución de estos objetivos se plantean las siguientes preguntas de investigación:

- ¿Qué caracteriza a los datos sucios en el ambiente empresarial cubano?
- ¿Modificar la distancia de edición, utilizando la cercanía de los caracteres en el teclado ayudará a que las técnicas de agrupamiento sean apropiadas para la estandarización de datos tipo cadenas de caracteres?
- ¿Será posible combinar las técnicas reportadas en la literatura sobre reemplazo de valores ausentes en un procedimiento que ayude a especialistas no expertos en estadísticas en esta tarea?

La **novedad científica** de la investigación está dada por:

- La definición de la distancia de edición entre cadenas EDUK (Edition Distance Using Keyboard) y su generalización d_{TOK}, en que se tienen en cuenta errores tipográficos. Además la definición de un marco de trabajo para resolver la estandarización de cadenas de caracteres y de forma particular de las direcciones postales urbanas, en que se utilice la distancia definida.
- La definición de un método para la sustitución de valores ausentes en campos de dominio discreto, y un procedimiento general que conduzca el análisis y sustitución de los valores ausentes.

Justificación de la investigación

La limpieza de datos como se ha mencionado anteriormente tiene múltiples aristas, por lo que el presente trabajo se enfoca en tres aspectos importantes: determinación de una taxonomía de errores en el ambiente cubano, la propuesta de un marco de trabajo para la

estandarización de cadenas de caracteres y la propuesta de un procedimiento que emplea las principales técnicas reportadas en la literatura para el reemplazo de valores ausentes, pero que permita su aplicación de forma asistida a un especialista no experto en estadística.

La determinación de la taxonomía brindará una caracterización del problema de los datos sucios en el entorno cubano, que por las características propias de desarrollo de la informática en el país, no tiene necesariamente que coincidir con las brindadas en la literatura especializada, además podrá aportar aspectos específicos sobre los cuales es necesario prestar atención en las tareas de limpieza.

La estandarización de cadenas se tratará en dos direcciones fundamentales: cuando las cadenas nombran objetos y cuando son direcciones postales.

Debido a la naturaleza de algunas cadenas, el proceso de entrada por el teclado no está sujeto a ninguna verificación, o sea, es libre. Es común encontrar cadenas que nombran al mismo objeto pero escritas de maneras diversas, por el uso de abreviaturas, errores tipográficos, etc. Es necesario someterlas a un proceso de estandarización, de manera que el mismo objeto siempre sea nombrado de la misma forma. Otras cadenas de entrada libre que se utilizan frecuentemente son las direcciones postales, que en Cuba se escriben de formas muy disímiles, no solamente porque se nombre a un mismo objeto de diferentes formas, sino por la propia estructura de la dirección, en que los elementos se escriben en cualquier orden. Para hacer uso de las mismas, es importante tener direcciones correctas, sin errores. Una forma de lograrlo es segmentándolas en diversas partes y después estandarizar cada una de esas partes como cadenas independientes.

Por todo lo anterior el trabajo que se propone tiene los siguientes valores:

Valor Práctico: Se formula una taxonomía de errores en el ambiente empresarial cubano y se obtienen herramientas de limpieza de datos que permita resolver los problemas de estandarización de cadenas y la sustitución de ausentes que se presentan comúnmente en sistemas que empleen grandes volúmenes de datos y que pueden ocurrir también en el proceso de extracción, transformación y carga de los Almacenes de Datos.

Valor Teórico: Se presentan procedimientos para la realización de las tareas de estandarización de cadenas y se propone una nueva distancia entre cadenas. Además un método para el reemplazo de valores ausentes de dominio discreto.

Valor Metodológico: Se sistematizan los conocimientos sobre limpieza de datos y establecimiento de procedimientos para la estandarización y reemplazo de valores ausentes, lo que ha quedado recogido en la Monografía "Limpieza de datos" [87] en la que se aborda la definición del proceso, su importancia, las áreas para las que constituye parte de su definición, los principales métodos, técnicas y herramientas utilizadas en el mismo.

Después de analizado el marco teórico se formularon las siguientes hipótesis:

- Hipótesis 1: "La utilización de una distancia de edición basada en la cercanía de los caracteres en el teclado mejora el desempeño del algoritmo de agrupamiento utilizado en la estandarización de datos tipo cadena".
- Hipótesis 2: "La utilización de un procedimiento que combine técnicas de reemplazo de valores ausentes permite realizar esta actividad por un usuario no especializado en esta problemática mediante el uso de un asistente interactivo".

El documento de la tesis ha sido estructurado en tres capítulos:

En el **primer capítulo** se presenta la definición de limpieza de datos y otras definiciones asociadas a este proceso como el concepto de anomalía o error en los datos, así como los principales métodos, técnicas y herramientas utilizadas en el mismo.

En el **segundo capítulo** se comienza con la presentación de la taxonomía de errores del entorno empresarial cubano y luego se describen las soluciones que se dan a los problemas de estandarización de cadenas de caracteres y el tratamiento de los valores ausentes.

El **tercer capítulo** se dedica a la descripción de las características de las herramientas obtenidas para la limpieza de datos y se muestran los resultados de la experimentación y las aplicaciones de las mismas a diferentes bases de datos.

Finalmente se incluyen las **conclusiones** que resaltan los principales resultados obtenidos en la investigación y las **recomendaciones** de aquellos aspectos a los que se considera deben dárseles continuidad.

1. LA LIMPIEZA DE DATOS

Este capítulo tiene como objetivo presentar el proceso de “limpieza de datos”, estudiar las principales definiciones asociadas a este, sus métodos y las herramientas empleadas para realizarlo, exponiendo una valoración crítica de las mismas.

En la era actual la informatización ha invadido casi todas las esferas de la sociedad, la economía, el gobierno y las investigaciones científicas. Esto ha llevado al desarrollo de sistemas de información que se caracterizan por el manejo de grandes volúmenes de datos. Tal acumulación de datos propicia la ocurrencia de anomalías e impurezas, fenómeno que distorsiona los resultados obtenidos de la interpretación y análisis de los datos, y provoca, como consecuencia, la elevación de los costos y la disminución de los beneficios de su procesamiento.

En investigaciones realizadas en la década del 90, se estimaba que la razón típica de existencia de anomalías e impurezas en los datos era de alrededor de un 5% [109, 118]. Estudios más recientes aseguran que entre el 40 y el 60 % de los datos coleccionados están sucios de una u otra forma [31, 135] y en [47] se hace referencia a que en la generalidad de las empresas, el 25 % de los datos críticos presentan errores cuya presencia frecuentemente se ignora, lo cual evidencia una correlación directa entre el crecimiento de la informatización, y la aparición de errores y anomalías en los datos. Para tratar de resolver tal problema, los estudiosos de estos temas, aplican diferentes métodos que intentan identificar y eliminar las anomalías e impurezas que puedan estar presentes en los datos a manejar.

1.1. Definición de error o anomalía en los datos.

La anomalía es una propiedad de los valores de los datos, que ocasiona una representación errónea del mini mundo que estos reflejan. Puede ser originada por mediciones erróneas, entradas de datos sin validación, omisiones ocurridas mientras se coleccionan o se mantienen los datos, también puede ser el resultado de malas interpretaciones del análisis de los datos o de cambios que se han producido en el mini mundo que no se han reflejado en la representación de los datos [115].

Los datos que contienen anomalías, se denominan datos erróneos o sucios y su presencia puede obstaculizar el uso efectivo y eficiente de la información. Por ejemplo, si una empresa utiliza una lista de clientes con sus direcciones, la aparición de direcciones erróneas será causa de un mal servicio; los clientes pudieran no recibir la notificación del recibo de pago con las consecuentes insatisfacciones y la disminución del prestigio de la empresa.

1.1.1. Clasificaciones de los errores o anomalías en los datos.

Para su estudio, las anomalías en los datos se organizan en las llamadas taxonomías de errores, de forma tal que se puedan identificar los problemas de calidad presentes. Son pocos los artículos que, desde diferentes puntos de vista, se refieren a este problema de la identificación y clasificación de errores.

Según Heiko Müller [104], las anomalías de los datos pueden clasificarse, en:

- anomalías sintácticas: se refieren a errores léxicos, errores de formatos y dominios, y errores de no estandarización de la información,

- anomalías semánticas: incluyen violaciones en las restricciones de integridad en las tuplas, contradicciones en valores de datos que violan algún tipo de dependencia entre ellos, tuplas duplicadas y tuplas no válidas y
- anomalías de contexto: incluyen la ausencia de valores de datos en las tuplas o ausencia de tuplas completas que existen en el universo y no han sido representadas.

Otra clasificación de las anomalías tiene en cuenta la cantidad de datos involucrados en las violaciones de las restricciones, que pueden ser: un solo valor en una tupla, múltiples valores en una tupla, valores en una o varias columnas de tuplas o conjuntos de tuplas de diferentes relaciones. También se distingue entre los errores o anomalías a nivel del esquema o a nivel de instancia, y si su existencia ocurre en una única fuente o es el resultado de la integración de múltiples fuentes [104, 115].

En [50] se ofrece una taxonomía de 26 tipos de errores, en la que se incluyen valores o datos ausentes, datos incorrectos en una fuente de datos y datos inconsistentes o ambiguos cuando se trata de varias fuentes de datos. El autor mismo reconoce que está incompleta.

Otras taxonomías más recientes son las definidas por Kim en [65] en que los errores se organizan, de una forma muy completa, en una jerarquía que se va desglosando a partir de los datos ausentes, datos presentes pero erróneos y datos presentes sin error pero no usables, y la taxonomía de Oliveira [107], en que se analizan y se incluyen los errores descritos en las anteriores y algunos nuevos, referidos a violaciones de restricciones de dominios.

1.1.2. La anomalía de valor ausente.

Como se puede observar, en todas las taxonomías y clasificaciones tratadas en la sección anterior, una anomalía que aparece en grandes conjuntos de datos muy frecuentemente, es la ausencia de valor en algunos de los campos de los registros de datos, fenómeno conocido en la literatura como valor ausente, en inglés missing value.

La ausencia de valores en los datos se debe a diferentes causas, las más comunes son: ausencia de respuesta del cliente (por ejemplo en una encuesta), fallas en la transcripción de datos, fallas en el soporte físico de los datos, mal funcionamiento de los sistemas de adquisición de datos, no aplicabilidad del valor del campo al registro de información, entre otras.

Existen dos formas diferentes de considerar la ausencia de información [112]:

- **Dato ausente**: es un valor que no está en el conjunto de datos; pero que existe en el mundo real, sencillamente por algún error no aparece en la base de datos operacional.
- **Dato nulo (o vacío)**: es un valor que está fuera de la definición de cualquier dominio, el cual permite dejar el valor del atributo "latente"; es un dato que falta, o sea, no existe en el mundo real. En otras palabras, un valor nulo no representa el valor cero, ni una cadena vacía, estos valores tienen significado; el valor nulo implica ausencia de información porque se desconoce el valor del atributo o simplemente para ese objeto no tiene sentido.

Un ejemplo que muestra la diferencia antes expuesta es el siguiente: En una empresa que vende bocaditos se controla qué salsa prefieren los clientes (existen dos tipos de salsa: ketchup y mostaza, que son excluyentes) y su sexo. Un usuario pide un bocadito sin salsa

y el operador del sistema olvida introducir el sexo del usuario. Habrá entonces dos campos sin valor: sexo y tipo de salsa, el primero existe (el usuario tiene un sexo determinado), sin embargo el segundo no existe (el usuario no quería ninguna salsa). Al tratar de imputar algún valor para estas ausencias se puede "sugerir" un dato para el sexo, no así para la salsa [36].

En los sistemas de bases de datos habitualmente no se hace diferencia entre estos dos tipos de ausencias, sencillamente se deja vacío el campo o en el mejor de los casos se utiliza el valor especial NULL. En el caso de los sistemas de bases de datos en nuestro país es común no utilizar la marca NULL y en su lugar se escribe 0 (si es un dato numérico) o una cadena vacía en el caso de cadenas. Esto trae una complejidad adicional en el tratamiento de los valores ausentes pues el cero y la cadena vacía pueden tener significados concretos diferentes a la ausencia de valor [92].

Los valores ausentes constituyen un problema muy frecuente en cualquier estudio que se haga con los datos. Cuando estos valores son menores que 1% generalmente se consideran triviales; de 1-5 %, manejables; de 5-15% se requieren métodos sofisticados para manejarlos y de más de un 15% afectan seriamente cualquier clase de interpretación [33].

En [82] se establece que es importante reconocer de qué manera se han producido las ausencias de valor y se clasifican los tipos de ausencias en Completamente Aleatoria (MCAR – Missing Completely At Random, cuando la probabilidad de ausencia del valor de una variable es completamente independiente del valor del mismo y de cualquier otra variable); Aleatoria (MAR – Missing At Random, la ausencia del valor de una variable no depende del valor mismo, pero sí del valor de alguna otra variable) y en No Ignorable

(cuando la ausencia está relacionada con la variable en sí misma y no es predecible desde ninguna otra variable del conjunto).

Los valores ausentes deben ser reemplazados al cargarse en el Almacén de Datos, pues su presencia influye negativamente en los procesos de toma de decisiones y los modelos que se obtienen de estos datos pueden no expresar la realidad correctamente.

El proceso de reemplazo de valores ausentes será referido en el presente trabajo como imputación de valores ausentes.

1.1.3. La anomalía de falta de estándares en la representación de los datos.

Un tipo de anomalía sintáctica, que se observa también como un fenómeno muy frecuente en las bases de datos, es la falta de estándares en la representación de la información.

En el mundo de los datos, un estándar es un modelo al cual deben responder todos los objetos de la misma clase. En bases de datos una representación estándar de un dato significa que su valor esté conformado de acuerdo a un formato preestablecido.

Los estándares deben ser definidos por la organización responsable del dato. Dicha organización debe tener autoridad para lograr que dichos formatos sean utilizados por todos los elementos de la entidad [93]. En el momento de confeccionar los sistemas de información de la organización deben asumirse estos formatos y en la medida de lo posible, los analistas de sistemas deben velar porque esto se haga.

Algunas de las causas de la falta de estándares en los datos pueden estar dadas por:

- No hacer un uso adecuado de las restricciones en el propio proceso de creación de la base de datos. Por ejemplo, si al definir el atributo “sexo” no se incluye la restricción de permitir solamente los caracteres “M” o “F” al entrar la información se pudiera

producir entradas para este campo tales como “m”, “masculino”, “varón”, “hembra”, “mujer” que representan el sexo de formas diferentes, pero además pudiera ocurrir que se introduzca cualquier otra cadena que no lo indique.

- No utilizar las posibilidades que brindan hoy los lenguajes de programación para diseñar e implementar la entrada de datos, por ejemplo, máscaras para entradas de datos, componentes visuales y controles de interfaces gráficas y notificaciones para indicar cómo debe realizarse la entrada del dato, etc.
- La no correspondencia en los estándares establecidos cuando los datos provienen de la integración de múltiples fuentes.

A pesar de que los esfuerzos que se puedan hacer en las etapas de análisis y diseño de los sistemas informativos de una organización, es muy difícil lograr la estandarización de los datos, en muchos casos por la ausencia de estándares para los datos, porque no han sido definidos o porque la propia naturaleza de los datos impide crearlos. Cuando no hay estándares establecidos, la captura de la información se hace de una forma “libre”, habitualmente se presenta un control que permite escribir prácticamente cualquier cosa [85].

El problema se agrava cuando la organización cuenta con sistemas de información que se han implementado sin tener en cuenta los estándares establecidos y es necesario utilizar la información que contienen en la toma de decisiones.

1.1.3.1. La estandarización de datos tipo cadenas

Uno de los problemas que se trata en el presente trabajo: la estandarización de campos tipo cadena está muy cerca de la corrección ortográfica, aunque no es exactamente igual.

El problema de estandarizar, como ya se ha dicho, consiste en encontrar cadenas escritas de diferentes formas, pero que se refieran al mismo objeto y reemplazarlas por una forma de representación única. A diferencia de la corrección ortográfica no se tiene certeza de cuáles son las cadenas correctas. A pesar de esta diferencia es posible utilizar en su solución, algunas de las ideas que se emplean en los correctores ortográficos.

En [22, 28, 31, 71, 124] se señala que los errores tipográficos más comunes son: inserción (teclear "opertación" por "operación"), eliminación (teclear "gito" por "grito"), sustitución (teclear "ruiso" por "ruido" y transposición (teclear "rodoe" por "rodeo"). A partir de esta idea, la distancia de edición entre dos cadenas se define como la cantidad mínima de inserciones, eliminaciones y sustituciones que deben hacerse para transformar una cadena en la otra.

En [124] se muestran las siguientes estadísticas que dan una idea sobre la frecuencia de estos errores:

Tabla 1.1 Estadísticas sobre los errores tipográficos más comunes.

	Documentos de la oficina del gobierno (EEUU)	Diccionario Webster
Transposición	2.6%	13.1%
Inserción	18.7%	20.3%
Eliminación	31.6%	34.4%
Sustitución	40%	26.9%
Total	92.9%	94.7%

En el trabajo [117] se hace un estudio de errores cometidos al teclear textos en idioma español y se observa que el más frecuente es la omisión de acentos, que en términos de operaciones de edición sería la sustitución de una vocal acentuada por otra no acentuada. También se reflejan como frecuentes aunque con menor prevalencia, la omisión de un

carácter, la inserción de un carácter, la sustitución y la transposición, lo que concuerda con el estudio antes citado.

Sin hacer un estudio estadístico detallado de esta problemática, se analizó cómo se comportaba esta situación en dos bases de datos reales cubanas. En una se tomó para el análisis el campo "apellido", recogido en la base de datos Censo sobre el uso de equipos electrodomésticos, a partir de una encuesta realizada en la provincia de Villa Clara y en la otra el atributo seleccionado fue "causa de baja" de la base de datos de los contribuyentes de la ONAT del municipio de Ranchuelo. La situación encontrada se puede observar en la tabla 1.2.

Tabla 1.2 Estadísticas de errores tipográficos en las bases de datos Censo y la ONAT[1].

	Censo	ONAT
Transposición	0.7%	6.1%
Inserción	3.1%	14.3%
Eliminación	6.3%	20.4%
Sustitución	85%	44.9%
Total	96%	85.7%

Se observa que los resultados son similares a los descritos en la literatura. El alto número de sustituciones que se aprecian en la base de datos del Censo, se debe fundamentalmente a la ausencia de acentos en palabras.

Las técnicas clásicas de recuperación de información en los gestores de datos utilizan operadores (igualdad, subcadena, parte de, incluido en, entre otros) que requieren la igualdad para recuperar. En todos estos casos se trata de alguna manera de buscar la cadena exacta.

[1] El total de los por cientos no es 100 pues hay errores de otro tipo que no son los que se describen en la tabla.

Con el objetivo de encontrar cadenas similares en las bases de datos, se reportan en la literatura varios trabajos [18, 27, 28, 44, 45, 59, 103, 111, 120, 121, 140] pero la mayoría de ellos se refieren al caso particular en que las cadenas son nombres de personas o de entidades, en inglés y otros idiomas.

La estandarización de campos de tipo cadenas es uno de los pasos necesarios en la limpieza de datos, en cualesquiera de los contextos en que ésta se haga. En particular, si la limpieza se hace en el proceso de Extracción, Transformación y Carga de un almacén de datos, cobra especial importancia por la necesidad de contar con información confiable para la toma de decisiones.

Hoy en día los sistemas de gestión de bases de datos comerciales y los diferentes paquetes para la construcción de los sistemas de información pueden contribuir, desde la entrada de datos, a la estandarización de los datos tipo cadena. La utilización de reglas de chequeo, disparadores, restricciones de integridad, etc. son herramientas que, bien empleadas por los diseñadores de bases de datos, permiten que los errores en los datos disminuyan. Por otro lado la utilización eficiente de los diferentes controles (cajas de listas, cuadros combinados, botones radiales, correctores ortográficos, etc.) que brindan los entornos de programación para la captura de datos también puede contribuir a obtener datos conforme a un estándar.

No obstante hay determinados datos, cuya entrada resulta muy difícil de controlar: nombres de personas, entidades, causas de fenómenos, direcciones postales, etc. En todos estos casos el proceso de entrada de datos es “libre” y surge el peligro potencial de usar dos o más cadenas diferentes para representar el mismo objeto.

1.2. El proceso de limpieza de datos

Una solución lógica al problema de eliminar los errores e inconsistencias en el nivel de instancia en conjuntos de datos ya existentes, es explorar el conjunto de datos hasta encontrar los posibles errores y, una vez detectados éstos, proceder a su corrección.

La realización manual de este proceso ha quedado descartada pues tendrían que ser empleadas una gran cantidad de horas-hombre, de ahí que la comunidad científica haya dedicado grandes esfuerzos al desarrollo de métodos y algoritmos computacionales que se encarguen de detectar y eliminar las anomalías e inconsistencias de los datos, los cuales han sido agrupados bajo el término de limpieza de datos [7, 55, 94, 96, 98, 108].

Sin embargo, no existe una descripción única sobre los objetivos y el alcance del proceso, y se aplica con una comprensión y demanda variables en diferentes áreas del procesamiento y mantenimiento de datos [95]. La definición de limpieza de datos depende del área en que este proceso se lleve a cabo.

La limpieza de datos, también llamada lavado de datos (data scrubbing), trata de detectar y eliminar los errores e inconsistencias en los datos con el objetivo de mejorar su calidad. Los problemas de calidad de los datos pueden estar presentes en colecciones de datos tales como ficheros y bases de datos, por causa de errores ortográficos en la entrada, información ausente o datos no válidos. Cuando se integran múltiples fuentes de datos, como ocurre en los almacenes de datos, los sistemas de bases de datos federadas y los sistemas de información basados en web, la necesidad de la limpieza crece significativamente [115].

Existen algunas áreas de trabajo de las bases de datos en que el proceso de limpieza se incluye como un componente propio, las principales son: los almacenes de datos DW,

(Datawarehouse), el descubrimiento de conocimiento en bases de datos KDD (*Knowledge* Data Discovery) y el manejo de la calidad de los datos TQMD (Total Quality Management Data).

1.2.1. La limpieza de datos en los almacenes de datos

Los almacenes de datos constantemente están cargando y refrescando cientos de datos de diferentes fuentes, por lo que la probabilidad de que algunas de las fuentes contengan datos sucios es alta. Además, como los almacenes de datos se usan en sistemas de ayuda a la toma de decisiones, la corrección de estos es vital para evitar llegar a conclusiones erróneas [58]. Por ejemplo, información duplicada o ausente podría producir estadísticas incorrectas; debido a esto, la limpieza de datos se considera uno de los mayores problemas en los almacenes de datos.

En este campo, la limpieza de datos se utiliza típicamente cuando se mezclan varias bases de datos. En este contexto, datos que representan la misma entidad, se expresan de formas diferentes en los distintos conjuntos de datos y, al mezclarse aparecen registros duplicados. El objetivo de la limpieza es detectar y eliminar los registros duplicados, lo que frecuentemente se conoce como el proceso de Merge/Purge [53, 55, 66]. Otras referencias a este problema son presentadas en la literatura como enlace de registros (Records Linkage), integración semántica, identificación de instancias, o el problema de la identidad del objeto [60, 126].

Para la eliminación de registros duplicados se han desarrollado varios métodos, entre ellos se relacionan los siguientes: limpieza de datos basada en el conocimiento [74], limpieza de datos basada en la correspondencia de expresiones regulares y restricciones

definidas por el usuario [15], limpieza de datos a través de un esquema de filtrado [131] y otros que se describen en [3, 27, 42, 75, 81].

En el contexto de los almacenes de datos, la limpieza se produce como un componente principal de la etapa de extracción, transformación y carga (ETL por sus siglas en inglés, Extract, Transform and Load) [9, 67, 137].

Las siguientes tareas describen en [115] el proceso de limpieza dentro del ETL:

a. **Análisis de datos:** Tarea para detectar los tipos de errores e inconsistencias que deben ser eliminadas.
b. **Determinación del flujo de trabajo para las transformaciones y las reglas de correspondencia:** Esto dependerá del número de fuentes de datos, su grado de heterogeneidad y la suciedad de los datos en ellas, de acuerdo con esto será el número de pasos de transformaciones y limpieza que serán ejecutados.
c. **Verificación:** El grado de corrección y efectividad del flujo de trabajo de las transformaciones, y de las definiciones de las transformaciones deberá ser probado y evaluado.
d. **Transformación:** La realización de los pasos de la transformación ya sea por la ejecución del flujo de trabajo del ETL para la carga o durante la ejecución de solicitudes a las diversas fuentes.
e. **Flujo inverso de los datos limpios:** Después de que los errores se eliminan, el dato limpio debe reemplazar al sucio en las fuentes para así mejorar su calidad y evitar rehacer el trabajo de limpieza en futuros procesos de extracción.

En ETL, la limpieza de datos es calificada como uno de los pasos que añade valor a los datos.

Aun cuando la eliminación de duplicados es un aspecto muy importante, sobre todo en los almacenes de datos, existen otras razones por las que un dato puede ser considerado sucio; debido a esto, la limpieza de datos se realiza generalmente de forma interactiva para que, en dependencia del conjunto de datos en específico, los expertos puedan establecer cómo enfrentara el problema definiendo las reglas que permitirán establecer la validez de los datos.

En esta dirección, en la literatura se reportan varias técnicas en que los usuarios definen las reglas y transformaciones para limpiar los datos, por ejemplo en [116] se propone una hoja de datos interactiva, a través de la cual el usuario puede establecer transformaciones basadas en restricciones definidas por él. Por otra parte en [42] se propone una interfaz SQL (Structured Query Language) para que los usuarios definan sus reglas y condiciones, en [25] se propone la definición de patrones de referencias para hacer corresponder los registros existentes, aplicando algoritmos de la lógica difusa y en [32] se propone utilizar las reglas del negocio para definir, en la fase de entrada, las restricciones a los datos.

Desde esta perspectiva, la limpieza de datos se define de varias formas que muestran similitudes. Por ejemplo, en [42] se define como el proceso de eliminación de errores e inconsistencias en los datos y la solución del problema de la identidad del objeto. Hernández y Stolfo [55] la definen como el problema de merge/purge y proponen el método básico de las vecindades ordenadas para resolverlo.

En [64] se define como el proceso de detectar y eliminar los conflictos en el nivel extensional cuando se integran dos o más fuentes de datos, y se propone una solución que se focaliza en la eliminación de conflictos en la terminología, basándose en el conocimiento lingüístico que se brinda a través de un dominio ontológico.

Kimball en [66] expresa que la necesidad de limpiar los datos constituye un problema universalmente conocido y, generalmente, ignorado; describe varios ejemplos de aplicaciones en que la calidad de los datos es un aspecto crítico, y plantea que la tecnología y el mercado del proceso de limpieza de datos están dirigidos fundamentalmente a la limpieza de listas de clientes, utilizando este ejemplo para describir la ciencia que subyace en la limpieza de datos. Manifiesta este autor que la limpieza de datos es mucho más que la actualización de registros con datos buenos, plantea que este proceso requiere de una descomposición y reensamble de los datos, y señala que el mismo debe seguir los siguientes pasos, particularizando para el caso de las listas de clientes:

- **Elementarización:** Separar los elementos del dato en partes lógicas con un significado.
- **Estandarización:** Transformar las partes a una forma estándar, por ejemplo, sustituir el uso de abreviaturas por términos completos.
- **Verificación:** Revisar la consistencia de lo estandarizado.
- **Correspondencia:** Chequear los datos que aparezcan en otros registros y que todos sus elementos sean idénticos.
- **Householding:** Analizar que si dos clientes tienen una misma dirección, deben aparecer en otra fuente como un matrimonio o tener algún parentesco.
- **Documentación:** Documentar los resultados en un metadato, asegurando que en otras aplicaciones del proceso los resultados de reconocer un cliente sean mejores.

1.2.2. La limpieza de datos en el área de Calidad de los Datos

En el área de TQMD, el proceso de limpieza de datos es de interés para la comunidad científica y de los negocios. La calidad de los datos y su integración a los procesos de negocios se reflejan en la literatura desde varios puntos de vista [20, 38, 79, 109, 118, 138].

Levitin y Redman [79] proponen un modelo para el ciclo de vida de los datos con aplicación en la calidad de los datos, estableciendo las actividades de evaluación, análisis, ajuste y descarga de datos para los ciclos de adquisición y uso de los datos. En este mismo marco de la calidad de los datos, Fox [38] propone cuatro dimensiones de calidad de los datos: precisión, actualidad, integralidad y consistencia. El grado de corrección de los datos está definido en función de estas dimensiones. Así, en este marco, el proceso de limpieza de datos es aquel que trata de "precisar el grado de corrección en los datos y mejorar su calidad".

1.2.3. La limpieza de datos en el proceso de KDD

Por otra parte en el área de KDD la **limpieza de datos** se define como el primer paso o preprocesamiento [14, 136]. Varios sistemas de KDD y minería de datos resuelven las actividades de limpieza de datos con herramientas dependientes de dominios específicos. Por ejemplo, en [51] se utilizan los patrones informativos para descubrir patrones erróneos en los datos. Las técnicas de aprendizaje automático (machine learning) también se aplican en el proceso de limpieza de datos para el problema de la clasificación de caracteres.

En la minería de datos se hace énfasis en el proceso de limpieza de datos, teniendo en cuenta el principio "Si basura entra, basura sale" y por otra parte, varias de las técnicas de

la propia minería de datos son utilizadas en el proceso de limpieza. La detección de extremos es un problema de especial interés, cuyo objetivo es encontrar excepciones en grandes conjuntos de datos, que posiblemente constituyen errores. Para resolver esto, se han propuesto diversas soluciones como por ejemplo: la determinación de extremos basada en la distancia [68], la aplicación de técnicas de agrupamiento (clustering) [142] y las redes neuronales [54].

A partir de lo anterior Maletic and Marcus, en el libro "Data minning and Knowledge discovery handbook" [98], definen que el proceso de limpieza de datos es aplicado directamente en las etapas de adquisición o definición de los datos, o después que estos pasos son realizados, para mejorar la calidad de los datos en un sistema existente.

Definen además tres pasos para realizarlo:

- Definir y determinar errores,
- Buscar e identificar instancias de errores,
- Corregir los errores descubiertos.

Cada una de estas fases constituye un problema complejo en el que se puede aplicar una gran diversidad de métodos y tecnologías especializados.

Cualquier enfoque para la limpieza de datos debe satisfacer los siguientes requerimientos:

1. El primero de todos es que a través de este proceso deben ser detectados y corregidos los principales errores e inconsistencias tanto para fuentes de datos simples o cuando se integran diversos orígenes de datos.
2. En segundo lugar, el enfoque debe estar basado en herramientas que limiten tanto la inspección manual de los datos como los esfuerzos de programación y permitan ser aplicadas fácilmente a otros orígenes de datos.

3. Además el proceso de limpieza de datos no debe resolverse aisladamente sino junto a las trasformaciones de los datos relativas a los esquemas, basándose éstas en los metadatos [115].

Para este trabajo se define la limpieza de datos como el proceso mediante el cual se detectan y corrigen los errores en los datos, proceso que es válido aplicar tanto en bases de datos de uso operacional como en aquellos conjuntos de datos que sirvan de fuentes a los almacenes de datos.

1.3. Análisis de datos

Independientemente del contexto en que se esté desarrollando la limpieza de datos, ya sea en el proceso de ETL de los almacenes de datos, en el preprocesamiento de los datos en el KDD, o en las aplicaciones de TQDM, un paso común a realizar es el análisis de datos. Este paso es fundamental ya que no se puede mejorar la calidad de los datos sin antes haber realizado un análisis detallado de los mismos [108].

Para realizar este análisis, un primer elemento a tener en cuenta es la revisión de los metadatos, que se crean en los gestores para guardar información sobre los datos, de este examen se pueden derivar importantes características que posibilitarán determinar su calidad. Sin embargo en muchas ocasiones los metadatos son insuficientes, más aún en bases de datos en que las restricciones de integridad no han sido establecidas o son muy pocas las que han sido definidas y aplicadas. En estos casos, un análisis de las instancias de datos puede contribuir a la obtención de nuevos metadatos que sirvan para determinar problemas en la calidad de los datos.

Existen dos enfoques muy interrelacionados para realizar el análisis de los datos [115]: el perfil de los datos (data profiling) y la minería de datos. El perfil de los datos, se centra en el análisis individual de las instancias de los valores de un atributo y la minería de datos, ayuda al descubrimiento de patrones en grandes conjuntos de datos.

1.3.1. El perfil de los datos

El perfil de los datos ha emergido como una nueva tecnología, que emplea métodos analíticos con el propósito de desarrollar la comprensión del contenido, estructura y calidad de los datos [108]. Con el perfil de los datos se obtiene información sobre el tipo, la longitud, el rango de valores, la frecuencia de los valores discretos, la varianza, la unicidad, la ocurrencia de valores ausentes, los patrones típicos en las cadenas, entre otros elementos que ofrecen una visión exacta sobre varios aspectos relacionados con la calidad de los datos del atributo analizado.

1.3.2. Minería de datos

En el análisis de datos son utilizados principalmente los modelos descriptivos de la minería de datos, que incluyen las técnicas de agrupamiento, y del descubrimiento de asociaciones y de secuencias [134]. A través de estas técnicas también pueden ser generados metadatos que posteriormente podrían ser usados para sustituir valores ausentes, corregir valores ilegales e identificar duplicados.

La detección de los errores y la determinación de las instancias en que estos ocurren, son problemas complejos y aunque el análisis de la integridad de los datos puede revelar algunos tipos de errores, existen otros muy difíciles de detectar como es el caso en que los errores involucran varios campos interrelacionados. Es aquí donde la detección del

error requiere de un profundo análisis. Un procedimiento general que permitiría descubrir este tipo de errores es detectar los datos que se comportan de forma diferente, o sea, determinar las excepciones del comportamiento general.

Por ejemplo, la aplicación de la técnica de minería por reglas de asociación ha sido utilizada para la detección de errores en los datos [16, 97], basándose en el hecho de que si una determinada regla es descubierta y se cumple para un alto por ciento de los datos, el 99% por ejemplo, el 1% restante debe ser examinado como error potencial.

1.4. Métodos generales usados en la limpieza de datos

Los métodos para llevar a cabo la limpieza de datos están estrechamente ligados al área en que se aplica y el paso del proceso que se esté realizando. Sin embargo, se destacan algunos métodos generales:

Para la detección de errores en [98] se señalan como muy usados *los métodos estadísticos* que aunque simples y rápidos, pueden generar muchos falsos positivos; *los métodos de agrupamiento basados en distancias*, cuya principal desventaja radica en la complejidad computacional; *los métodos basados en patrones y en reglas de asociación* que, a partir del análisis de los registros que incumplen los patrones y las reglas descubiertas, detectan posibles errores.

También en [104] se describen el *parsing* [116], *las transformaciones a nivel de esquemas e instancias* [125], *el reforzamiento de las restricciones de integridad* [132], *el método de las vecindades ordenadas* [55, 75] y otros [5, 8, 12, 76, 102] que utilizan diferentes enfoques para realizar la eliminación de duplicados.

1.4.1. Métodos empleados en el tratamiento de los valores ausentes

En [141] se presentan diferentes métodos para imputar valores ausentes como parte del preprocesamiento de los datos que es necesario hacer para aplicar técnicas de minería de datos. Algunas de las técnicas descritas en [123] imputan los valores ausentes, considerando cada atributo sin relación con los demás. Otras técnicas son más complejas, utilizan la regresión, las redes bayesianas, los árboles de decisión y otros métodos que son descritos en [21, 36, 49, 57, 73, 82]. La mayoría de las técnicas asumen que la ausencia de valor es del tipo MAR o MCAR.

En [112] también se aborda el problema del reemplazo de valores ausentes. Se plantea la necesidad de encontrar los patrones de valores ausentes, que son n-uplas de valores binarios donde un uno representa presencia del valor y cero la ausencia del mismo. No queda claro en el libro la utilización posterior de estos patrones.

Como característica importante de los métodos de imputación se señala que no deben producir sesgo (unbiased estimators), de forma que no cambien las características más importantes de los valores presentes, cuando son incluidos los valores imputados. En particular se hace alusión a mecanismos que preserven la media, la desviación típica de los datos, mecanismos de interpolación para preservar la relación entre las variables (si esta relación es lineal) u otros mecanismos como las redes neuronales artificiales, si se conoce que la relación entre las variables es no lineal.

Un conjunto de datos con las variables Y_1, Y_2, ..., Y_p (en ese orden) se dice que tiene un patrón de valores ausentes monótono; si ocurre que una variable Y_j tiene un valor ausente para un caso particular, entonces para ese caso particular el resto de las variables Y_k, $k > j$, también tienen valores ausentes [61]. En este propio trabajo se utiliza esta propiedad para

determinar la conveniencia de la utilización de la regresión como método de imputación de estos valores y de otras técnicas como el método MCMC (Markov Chain Monte Carlos) [127] para convertir los patrones en monótonos y luego aplicar la regresión.

1.4.2. Métodos para la estandarización de cadenas de textos

El proceso de limpieza de datos parte del hecho de que en la base de datos existen datos "sucios", entre ellos, datos no estandarizados. Intentar estandarizar estos datos tipo cadena, conduce a encontrar cuáles cadenas son similares para que sean sustituidas por la correcta.

Con el objetivo de encontrar cadenas similares en las bases de datos, se reportan en la literatura varios trabajos [18, 27, 28, 43, 45, 59, 103, 111, 120, 121, 140] pero la mayoría de los trabajos consultados se refieren al caso particular en que las cadenas son nombres de personas, de entidades, en inglés y otros idiomas.

Para encontrar cadenas similares se utilizan funciones que generalmente constituyen distancias y en algunos casos métricas.

En la literatura se reportan varias formas de calcular la distancia entre dos cadenas [2, 28, 34, 111, 120]. En [35] se distinguen tres tipos de métricas: *las basadas en caracteres* (entre ellas la distancia de edición, la de Smith Waterman, la de Jaro, Q-Gram, etc), *las basadas en tokens* y *las basadas en sonidos* (Soundex, Metaphone, etc). Los trabajos [28, 106] constituyen referentes importantes para el estudio del problema de la aproximación entre cadenas.

En los gestores de datos el intento más generalizado de buscar cadenas similares (no exactas) está relacionado con la utilización de la función SOUNDEX (cadena) [44]. Esta

función asigna a una cadena dada un código y en lugar de comparar cadenas, se comparan dichos códigos, suponiendo que cadenas que "suenen" igual, tendrán el mismo código. Esta comparación tiene sus desventajas a partir del propio algoritmo que utiliza. Se han realizado intentos de mejoras como la función Phonix [39] y Metaphone [110], incorporadas también a algunos gestores, pero de cualquier manera la cantidad de registros recuperados en una consulta es baja, sobre todo en aquellos casos en que las cadenas son largas. Además, los algoritmos utilizados en los gestores utilizan fundamentalmente la fonética del idioma inglés.

La distancia de edición entre dos cadenas fue introducida por Levenshtein [78] como la cantidad mínima de operaciones de inserción, eliminación y sustitución que hay que hacer para transformar una cadena en la otra. La función así definida se demuestra que constituye una métrica y ha tenido algunas variaciones en el tiempo. Levenshtein consideró todas las operaciones con costo unitario, en trabajos posteriores se planteó que era posible que cada una de las operaciones de edición tuvieran costos diferentes [52], se adicionó una nueva operación: la transposición, intercambio de caracteres adyacentes que es un error frecuente en las personas que teclean rápido [143]. No se han encontrado en la literatura heurísticas para determinar los costos de las operaciones, aunque en [128, 130] se señala que, a partir de experimentos, se obtienen buenos resultados cuando los costos no son unitarios sino cuando tienen valor 2. En el presente trabajo se adoptará el nombre de distancia de Levenshtein-Damerau ($dist_{LD}$) al conjunto de estas métricas para tomarlas como referencia, por ser ellos los pioneros en su definición y constituir un referente de comparación.

La métrica de Smith Waterman [130], utilizada inicialmente para buscar alineación entre moléculas, utiliza la idea de la distancia de edición, y propone utilizar costos negativos en las operaciones de inserción y eliminación, y costos positivos para las operaciones de sustitución y para cuando haya coincidencia. Aquí se toma el costo de las operaciones de inserción y eliminación en función del tamaño del "hueco" que se produce al hacer una de estas operaciones.

La métrica de Jaro [59] es usada comúnmente para buscar similitudes entre nombres, en sistemas de enlaces de registros [139]. Se basa en el conteo de los caracteres comunes de ambas cadenas, el número de transposiciones y el empleo de una expresión donde intervienen estos valores y la longitud de las cadenas.

La métrica Q-Gram [133] y algunas de sus variantes [80] se basan en la determinación de subcadenas de longitud q (habitualmente se usa q igual a uno, dos o tres). Se buscan las subcadenas comunes en las dos cadenas pues se plantea que, si dos cadenas son similares, comparten un alto número de q-gram.

Las métricas basadas en tokens dividen las cadenas en partes (tokens), a partir de espacios en blanco, signos de puntuación, etc. Entre las métricas basadas en tokens se encuentra la "Similitud de Monge-Elkan" [101], que se calcula como la similitud máxima promedio entre una pareja de tokens. Una variante se presenta en [48] donde se utiliza la media aritmética generalizada en lugar del promedio. Otra métrica basada en tokens es la "similaridad del coseno" [18], en la que a cada palabra se da un peso en dependencia de la frecuencia relativa de aparición en la cadena, formándose vectores con estos valores y buscándose el coseno del ángulo que existen entre estos vectores. Estas técnicas son muy utilizadas en la recuperación de la información [26, 119].

Estas distancias, incluyendo las de edición, no tienen en cuenta de manera explícita los errores tipográficos. Considerando que la operación más frecuente dentro de los errores tipográficos es la sustitución, pudiera mejorarse el proceso de estandarización de cadenas de caracteres teniendo en cuenta la posición de los caracteres en el teclado al calcular la distancia entre dos cadenas.

1.5. Herramientas para la realización de la limpieza de datos.

Existen muchas herramientas para realizar la limpieza de datos, en [29, 30, 40, 56, 72, 105] se hace referencia a estas y se clasifican de acuerdo al tipo de tarea que resuelven dentro del proceso de limpieza de datos. En general son herramientas costosas, en la mayoría de los casos, las mismas compañías que las desarrollan, brindan el servicio de limpieza y además consideran elementos del entorno como son diccionarios de ortografía, geográficos, directorios y archivos de nombres locales.

Se hizo un estudio de las mismas y en la tabla 1.3 se recogen algunas de las más actuales y difundidas, señalando sus principales funciones y tareas que realizan, para lo cual se tuvieron en cuenta los aspectos de la limpieza de datos que son desarrollados en este trabajo: el análisis de datos, la estandarización de cadenas (incluyendo direcciones postales) y el reemplazo de valores ausentes.

Tabla 1.3 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, Detecta: dominios, llaves, dependencias funcionales, llaves extranjeras. Detecta patrones de datos como expresiones regulares. No reglas de asociación	Operador Match/Merge Busca tuplas similares y las mezcla en una. Puede ser por nombre, dirección o cadena. Utiliza la similitud entre cadenas para buscar similares. Distancia de Edición y Jaro-Winkler	Operador Name Address Necesita de licencia e instalación de software producido por terceros con información de nombres y direcciones para limpiar y estandarizar Cuba no aparece en sus listas	No
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: ausentes, vacíos por tipo de campo Numéricos: mínimo, máximo, suma, media, desviación, patrones. Cadenas: longitud	No	No	No

	máxima, mínima, cantidad de caracteres promedio. Valores más frecuentes			
SQL Server 2008	Perfiles de datos muy completos. Estadísticos de los campos numéricos, porcentaje de valores ausentes. 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

Como se observa en la tabla las herramientas realizan el análisis de los datos, usando la técnica de los perfiles, determinando los elementos más generales de los mismos. En cuanto a la estandarización de cadenas de caracteres solamente Oracle y SQL Server (ambas propietarias) tienen alguna funcionalidad relacionada con esta temática, pero son eficientes para dominios específicos. También estas dos herramientas poseen operadores para trabajar con direcciones postales, pero sin segmentarlas. El reemplazo de valores ausentes se hace utilizando los estadígrafos comunes sin brindar al usuario información adicional sobre ellos.

En el Anexo A se muestra otro conjunto de herramientas que se utilizan para efectuar la limpieza de datos. (Véase Anexo A).

1.6. Conclusiones parciales

La limpieza de datos se aplica con diferentes intenciones y dentro de diferentes áreas del proceso de manejo e integración de datos. Se ha definido en el presente trabajo como el proceso mediante el cual se detectan y corrigen los errores en los datos, que es válido

aplicar tanto en bases de datos de uso operacional como en aquellos conjuntos de datos que sirvan de fuentes a los almacenes de datos y se ha demostrado que es muy necesario debido al uso intensivo de grandes volúmenes de datos en los sistemas informativos actuales.

Los métodos y herramientas producidos para automatizar las tareas de la limpieza de datos, en general, están dirigidos a una tarea específica y en el caso de las herramientas, la mayoría son propietarias y están destinadas a resolver problemas en determinado dominio de aplicación, usando características locales.

En la revisión bibliográfica sobre el tema, se observó que ha sido poco abordado en las investigaciones en nuestro país; no existen productos cubanos de análisis y limpieza que tengan en consideración los problemas de calidad de los datos que se presentan en las bases de datos de los sistemas de información que se están desarrollando cada vez más. Es por eso que se justifica el desarrollo de nuevas taxonomías, métodos y herramientas que ayuden en el mejoramiento de la calidad de los datos de los sistemas informativos de nuestro entorno empresarial y de servicios.

2. DESCRIPCIÓN DE LAS SOLUCIONES PROPUESTAS PARA LA LIMPIEZA DE DATOS

En el presente capítulo se analizan las técnicas de limpieza de datos que, como resultado del estudio realizado, se han implementado.

Se presenta una taxonomía de errores para nuestro entorno empresarial y se exponen las técnicas utilizadas en la estandarización de los datos y en el tratamiento de los valores ausentes.

2.1. Taxonomía de Errores

La limpieza de datos es el proceso mediante el cual se realizan transformaciones a los datos para que adquieran la calidad que les permita aportar información confiable, a fin de contribuir a la eficiencia y efectividad de los procesos de toma de decisiones.

En la provincia de Villa Clara, algunas empresas confrontan el problema de contar con grandes volúmenes de información en los que la calidad de los datos no está garantizada:

- Banco Popular de Ahorro. En la dirección provincial de esta entidad, existen miles y miles de registros que almacenan la información sobre los clientes y las cuentas. Por ejemplo, al referirse al nombre de una calle en el campo dirección de una de las tablas de una base de datos, este se registra como S. Miguel o San Migel, o que en el campo nombre de la sucursal para referirse a la misma sucursal aparezca en ocasiones como Suc. 4292 y otras como Sucursal 4292, o sea, hay suciedad en los datos registrados de entradas erróneas e inconsistencias.

- La gerencia territorial de ETECSA controla e interactúa con una larga lista de clientes cuyas direcciones postales no presentan un formato uniforme, así las direcciones están conformadas por una estructura implícita, comprendiendo elementos como “calle”, “ciudad” y “código postal”; sin embargo, el orden de estos elementos no es fijo, no siempre todos los elementos están presentes en todas las direcciones y además el formato de las direcciones difiere dentro una misma provincia.

- La delegación del MINAZ posee grandes bases de datos en las que se almacenan todos los datos de las zafras y ocurren, por ejemplo, errores como valores repetidos, diferentes códigos con el mismo significado y valores ausentes.

- En los centros hospitalarios ha existido una tendencia al crecimiento de la automatización de sus procesos y sus bases de datos, que contienen información sobre pacientes, medicamentos, diagnósticos, personal médico, también presentan problemas de errores e inconsistencias en los datos.

Todo lo anterior exige realizar el proceso de “limpieza de los datos”.

Como se explicó en el capítulo 1 y en [108], obtener datos de calidad o limpiar los datos es un proceso que está ligado al hecho de cómo los datos son usados. En la literatura especializada se ofrecen varias taxonomías de errores que deben ser abordadas por los procesos de limpieza de datos y tenidas en cuenta en el momento del diseño de los sistemas para asegurar la calidad de los datos que se manejan.

El entorno empresarial cubano no está ajeno a esta problemática de datos sucios y, si se desea emprender la construcción de almacenes de datos que permitan la toma de decisiones acertadas, un problema que deberá ser tratado es la limpieza de datos. Aunque

los reportes científicos o de aplicación de tales intentos en nuestro país son muy escasos, se conoce de varios procesos de creación de almacenes en que el proceso de limpieza ha tomado una gran cantidad de tiempo.

Para esta investigación es importante corroborar que las taxonomías de errores que se señalan en la literatura sean aplicables al entorno cubano y, por lo tanto, que al enfrentarse al problema de limpieza de datos este pueda encaminarse a partir de dichas taxonomías.

Aunque los problemas que se señalan en las taxonomías son lo suficientemente generales, nuestro entorno tiene algunas características que lo particularizan y que podrían influir en la presencia o no de determinados errores.

Estos elementos particulares son los siguientes:

- Los sistemas de bases de datos han proliferado en la empresa cubana a partir de la aparición de las microcomputadoras. No quedan sistemas que corran en computadoras grandes (las que fueron originalmente del sistema SUMCE), ni tan siquiera en minicomputadoras (francesas - IRIS, cubanas - CID u otras),
- La característica anterior hace que se pueda afirmar que todos los sistemas de bases de datos se han concebido a partir del modelo relacional, o al menos se han concebido a partir de la utilización de algún gestor de datos que soporta dicho modelo,
- Los gestores de datos más utilizados a partir de la década del 90, son aquellos que soportan la tecnología xBase: FoxPro (para DOS y Windows), Visual FoxPro. Luego se extendió el uso de Access como gestor de datos y en años más recientes se viene usando el SQL Server en las diferentes versiones que han aparecido en el mercado.

Hay instituciones y organismos que han utilizado de manera esporádica algún otro gestor como Oracle, Sysbase, Informix, etc. Más recientemente se apuesta a los gestores denominados libres como MySQL o PostgreSQL. Todos estos gestores son de la familia de gestores relacionales, algunos con más soporte de los elementos del modelo que otros, pero todos tienen como elemento primario la tabla (relación),

- El personal que ha diseñado los sistemas de bases de datos no siempre ha sido el más idóneo y no conoce o no aplica en toda su potencialidad el modelo relacional. En particular, se menosprecia el importante papel de las restricciones de integridad en la calidad de los datos y muchas veces se deja este chequeo para los sistemas y no para la base de datos,
- Otro soporte de datos extendido en nuestras empresas son las hojas de cálculo, utilizando para ello mayoritariamente el Excel del paquete Microsoft Office.

Para llegar a una taxonomía propia de nuestro entorno, se diseñó un software denominado DBAnalyzer que, utilizando la técnica de los perfiles de datos (explicada en el capítulo 1), permitió el estudio de varias bases de datos, tarea no trivial pues significaba "cuestionar" los datos y su grado de confiabilidad, y con ayuda de analistas – usuarios de esas bases de datos se logró encontrar múltiples errores.

Las bases de datos estudiados fueron:

- Control de Zafra (dos instancias de zafras de finales del 90) (Zafra-1, Zafra-2)
- Control de Estudiantes de la UCLV (Est)
- Datos del BPA, Villa Clara (BPA)
- Control de Contribuyentes de la ONAT Ranchuelo (ONAT)
- Control de Defunciones (dos instancias, años 1999 y 2003) (Defu99, Defu03)

Un resumen de los datos analizados se muestra en la tabla 2.1.

Tabla 2.1 Características de las bases de datos utilizadas

Base de Datos	Cantidad de tablas	Cantidad de registros
Zafra-1	406	15620
Zafra-2	169	12608
Est	23	6786
BPA	17	88976
ONAT	9	3410
Defu99	2	79942
Defu03	2	78957

El análisis de estas bases de datos aportó elementos interesantes para el perfeccionamiento de la herramienta. Por ejemplo, se comprobó que es importante en cualquier herramienta de análisis de datos para las empresas cubanas poder analizar el atributo “Carné de Identidad”, utilizado en muchos casos como clave primaria cuando se trata de personas. Se encontraron violaciones de la unicidad en este atributo y valores que no corresponden al patrón de 11 dígitos, establecido para el mismo. Esta evidencia hizo que se incorporara una nueva verificación a la herramienta.

A partir de los resultados también son evidentes algunos de los criterios que se tenía a priori: muchos de los errores detectados se deben a errores de diseño de la base de datos y a la no utilización de las restricciones de integridad, a las que el modelo relacional da tanta importancia.

En [99] se muestran algunos resultados obtenidos en el análisis de las bases de datos de zafra, defu99, defu03 y Est; en [92] se detallan algunos de los problemas detectados en los datos del BPA; en [91] se muestran algunos resultados de la aplicación de la

herramienta a la Base de datos Estudiantes de la UCLV y en [84] se analizó el comportamiento de los valores ausentes en los datos de una empresa cubana.

La taxonomía de errores que se propone a partir del análisis, es la siguiente [88]:

Datos Incompletos

- ***Registros que faltan (tablas vacías)***: es el error que aparece cuando, realizándose un buen diseño de la base de datos, se definen las estructuras de las tablas y, por alguna razón, no se almacena en ellas información.

- *Campos que faltan:* generalmente se encuentra que las tablas de las bases de datos se diseñan con muchos campos, de los cuales se utilizan muy pocos; entonces la tendencia posterior es dejar vacíos esos que no son utilizados en el procesamiento.

Un ejemplo de lo anterior quedó contabilizado en la tabla 2.1.

Tabla 2.1 Falta de información en las bases de datos estudiadas

Base de Datos	Registros que faltan (Tablas vacías)	Campos que faltan[1]
Zafra-1	131	6
Zafra-2	18	0
Est	2	0
BPA	3	0
ONAT	0	0
Def-2	0	10

[1] Se considera campo vacío, a aquel campo que está vacío en todos los registros de la tabla.

Datos Incorrectos

- *Códigos incorrectos:* se utiliza un código que no aparece en el codificador asociado a ese atributo. Esta situación, a pesar de que el modelo relacional la soluciona con la ayuda de llaves extranjeras, se presenta de manera reiterativa en las bases de datos analizadas.

- *Registros duplicados:* información repetida en varios registros de datos que corresponde a un mismo objeto.

- *Información incorrecta entrada al sistema:* es el error cometido en el proceso de entrada, por fallas tipográficas o de otro tipo.

- *Carné de Identidad incorrecto:* por el papel que juega este atributo en las bases de datos (muchas veces llave primaria – llave extranjera) se considera como un error importante; los errores vienen dados por: no contar con 11 dígitos, no seguir al inicio el patrón: año, mes, día.

Datos Incomprensibles

- *Múltiples campos dentro de uno:* en un campo se escriben valores de atributos diferentes, separados por coma o algún otro signo o símbolo.

Datos Inconsistentes

- *Uso y significado inconsistente de diversos códigos:* en el transcurso del tiempo cambian, por alguna razón, los codificadores y existen en la base de datos referencia a los codificadores antiguos, también se puede dar el caso de que se utilicen distintos codificadores en bases de datos diferentes que posteriormente se hace necesario unificar.

- *Diferentes códigos con el mismo significado:* una entidad se codifica de varias maneras. Este error puede suceder en una única fuente de datos o en varias fuentes de

datos. Es más frecuente en el caso de grandes codificadores o de codificadores que tienen una naturaleza jerárquica, en que no se ha precisado el camino al que pertenece la entidad en la jerarquía.

- *Uso inconsistente de nulos, vacíos y espacios:* en el modelo relacional existe el **null** como marca para indicar ausencia de información, es muy común que en lugar de utilizar dicha marca, se utilicen cadenas vacías, espacios, valor cero y valores por defecto para indicar la ausencia de información.

- *Falta de integridad referencial:* no existe una correspondencia entre una llave extranjera y la llave primaria que refiere. Aparece este error cuando no se ha realizado un análisis adecuado de las llaves y por lo tanto no se han incorporado al esquema de la base, o cuando no se ha impuesto el chequeo correspondiente por el gestor de datos.

A partir de la taxonomía de errores, se comprobó que dos de los problemas que se presentan con mayor frecuencia en los datos, son la ausencia de valor y la falta de estándares en campos que aparecen en las formas de entrada con un formato libre, ya sea por errores ortográficos o de tecleo.

2.2. Solución propuesta para la estandarización de los campos tipo cadena de caracteres.

Una de las razones de la existencia de datos "sucios" en una base de datos es la escritura incorrecta en el momento de entrar los datos. La escritura incorrecta se puede producir por muchas razones:

- Errores ortográficos (en español, por ejemplo se cambia frecuentemente la "b" por "v", "s" por "c" o viceversa, se omite la letra "h", etc.).

- Errores por oprimir teclas incorrectamente (por ejemplo, si es necesario teclear la letra “a”, puede ser oprimida la letra “q” porque ambas están próximas en el teclado.
- Errores de sonido (alguien dice “maría” y la persona que está entrando los datos oye “manía”, también es frecuente que por una percepción errónea se omitan sonidos como “s” y “r” al final de las palabras.

Por estas razones aparece la estandarización como una etapa en el proceso de limpieza, a fin de lograr uniformidad en los datos.

El proceso de estandarización es un paso importante, previo a la detección de registros duplicados. Si se cuenta con los campos estandarizados, la detección de duplicados se hará de forma más efectiva, independientemente del método que se utilice para realizarla.

Marco de Trabajo.

La estandarización de datos es un proceso complejo y en el presente trabajo solo se han tenido en cuenta los datos de tipo texto.

Se propone a continuación un marco de trabajo que realiza la estandarización de un campo tipo cadena. Este marco de trabajo tiene la característica de ejecutarse en interacción con el usuario-analista de los datos, no pretende ser completamente automático, pues la tarea de estandarizar cadenas depende mucho del entorno en que se realiza y por tanto el conocimiento que se tenga de los datos, influirá de manera notable en el éxito de la estandarización. Está compuesto por las siguientes etapas:

Etapa 1: Identificar el atributo a estandarizar

Es un paso trivial, es necesario encontrar el atributo que se quiere estandarizar y comprobar la naturaleza tipo texto del mismo. Aquí además es necesario determinar si el atributo es una dirección, pues este tipo de atributo tendrá un tratamiento particular.

Etapa 2: Realizar sustituciones

Se ha comprobado que uno de los problemas fundamentales que presentan los campos textuales que pueden ser entrados de forma "libre" en un sistema informativo, es la utilización de algún tipo de abreviatura, o apócopes, que pueden, incluso, no ser los mismos a lo largo del proceso de entrada de los datos. Por ejemplo, para "avenida" se encuentran "ave", "ave.", "aven", "avn", etc. Otro ejemplo pudieran ser los apellidos, para "Fernández" es común encontrar "fdez" "fern", "fndez", etc.

Para realizar las sustituciones se define la función **sust[l, e],** donde **l** es una lista con las cadenas a sustituir y **e** es la cadena por la cual se sustituyen todos los elementos de la lista. Así, por ejemplo, sust[{"fdez", "fern", "fndez"}, "Fernández"], indica que hay que sustituir las cadenas que están en la lista por la cadena "Fernández".

Con estas sustituciones comienza el proceso de estandarización de las cadenas de texto, además de garantizar que los pasos que siguen tengan mejores resultados.

Por otra parte es necesario ir acumulando el conocimiento de estas sustituciones, por dos vías: incrementado la lista con las cadenas a sustituir a partir de lo encontrado en la práctica e ir aumentando el número de instancias de la función a partir de nuevas sustituciones que aparezcan en el proceso de estandarización.

Etapa 3: Utilizar el proceso de estandarización de direcciones si el atributo es de ese tipo

Un tipo especial de atributo, que se utiliza mucho en las organizaciones, son las direcciones postales. Contar con direcciones fiables, es necesario para muchos procesos. Estas pueden estar presentes en la toma de decisiones, cuando tienen que ver con la ubicación geográfica de determinados procesos.

Por lo general, en nuestro entorno las direcciones postales se capturan de manera única, quiere decir esto que en un mismo atributo se ubican todos los elementos de la dirección (calle, número, entre calles, apartamento, reparto, etc.). En algunos casos se separan la ciudad, provincia, pero el resto de los elementos están mezclados. La forma de escribir estos elementos es muy variada, lo que no permite su utilización efectiva, para ello es importante separarlos en diferentes campos. Se ha establecido un proceso que se ha llamado "estandarización de direcciones postales", a partir, fundamentalmente, de la utilización de los modelos ocultos de Markov, que permite realizar esta división de la dirección postal en elementos particulares. Es necesario destacar que la entrada al proceso es una dirección postal y la salida es la dirección segmentada en los distintos elementos componentes.

Etapa 3.a. Si el atributo textual a estandarizar no es una dirección postal se realiza el siguiente proceso:

Si el atributo texto no es una dirección postal se aplica el método PAM (Partition Around Medoides), introducido por Kaufman y Rousseeuw [63], para formar grupos con los valores del atributo. Para la utilización de este método de agrupamiento, la distancia de dos elementos se determina empleando una modificación de la distancia de edición de

Levenshtein, de tal manera que se tenga en cuenta la distancia en el teclado de los diferentes caracteres. (Este aspecto se explicará en el epígrafe 2.2.1). Con este agrupamiento se logra que cadenas similares queden reunidas en el mismo grupo.

Etapa 4: Reemplazar patrones encontrados

Si el atributo a estandarizar es una dirección postal, en este paso no hay nada que hacer, pues el proceso descrito anteriormente tiene como salida los elementos de la dirección que se determinaron.

Si el atributo es un texto cualquiera, se debe analizar los diferentes grupos obtenidos en la etapa anterior. El proceso ahora se reduce a la cantidad de grupos definidos, pues en cada uno de ellos se encuentran las cadenas más semejantes. Es necesario determinar, de forma interactiva entre analista y sistema, las sustituciones a realizar.

2.2.1. Nueva distancia de edición

En el marco de trabajo propuesto, se utiliza la idea de realizar un agrupamiento de las cadenas a estandarizar. Para realizar este agrupamiento, cualquiera que sea la técnica que se utilice, es necesario definir una distancia entre las cadenas. Debido a que los datos a estandarizar, provienen fundamentalmente de entradas, por teclado, se intuye que una distancia de edición dará buenos resultados en el agrupamiento y como se ha dicho que la forma de calcular los costos de las operaciones de edición es objeto de investigación hoy en día [11], se propone en el presente trabajo una forma de determinar el costo de la operación de sustitución, que es el error más frecuente de los errores de edición. En [86] se definen los primeros elementos asociados a esta distancia.

Para buscar este costo se construye un grafo con el teclado de la computadora G(V,A), donde V es el conjunto de teclas (vértices del grafo) y A es el conjunto de aristas. El vértice V_i está conectado con el vértice V_j, si representan teclas adyacentes en el teclado.

Una porción de este grafo se muestra en la figura 2.1:

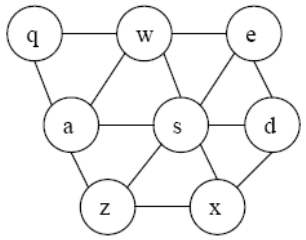


Figura 2.1

De esta forma el costo w_s de sustituir el carácter i por el carácter j, es la longitud del camino más corto para ir de i a j en el grafo G.

Al definir el costo de la sustitución de esta forma, se asegura que se tenga en cuenta la sustitución de teclear, en lugar de un carácter, otro adyacente al mismo. Esto es más probable que teclear un carácter más alejado. Por lo tanto debe disminuir la distancia entre las dos cadenas. También, y de una manera indirecta, se está asegurando "penalizar" de manera menos rigurosa algunas faltas de ortografía comunes en el idioma español, como por ejemplo, cambio de "b" por "v", "s" por "z", incluso "g" por "j", "s" o "z" por "c", pues son letras que en el teclado tipo QWERTY (los más usados en Cuba) son vecinas o casi vecinas.

De esta forma, por ejemplo:

d(perez, peres)=1

d(perez, pereo)=2

En el segundo caso la distancia es dos pues en lugar de hacer la sustitución de “z” por “o” (tendría peso ocho) se hace una eliminación y una inserción (con peso uno cada una). Se puede observar que si hubiera utilizado el mismo costo para todas las sustituciones, ambas distancias serían igual a uno. En este caso, el par (perez, pereo) se está distanciando más, lo que puede indicar que no es muy lógico considerar a “pereo” como un error de “perez”, sino de alguna otra cadena (por ejemplo de “peredo”). Esto es importante en el momento de hacer el agrupamiento, al tener una distancia mayor, es mayor la probabilidad de que las cadenas caigan en grupos diferentes.

En esta forma de definir el costo de la sustitución, el teclado juega un papel importante, aunque la parte que cambia más en los teclado QWERTY son los signos de puntuación que están a la derecha del mismo, u otros símbolos que aparecen a la izquierda de la “z” o del número “1”. Estos símbolos son menos usados en las cadenas y se considera que no son un problema para la definición que se ha dado. Otro problema que se presenta en español es el uso de las tildes. Se hace la siguiente consideración: las vocales con tilde se consideran vecinas de las vocales sin tilde y vecinas de todos los caracteres vecinos de la vocal sin tilde.

Formalizando estos resultados, se define el cálculo de la nueva distancia de edición entre las cadenas s y t de longitud n y m respectivamente, la cual se llamará en lo adelante EDUK (Edition Distance Using Keyboard), a través de una matriz de dimensión (n x m) de forma tal que:

$d(i, 0) = i, i = 0, 1, ..., n$

$d(0, j) = j, j = 0, 1, ..., m$

Para el resto de los valores:

$$d(i,j) = \min \begin{cases} d(i-1, j) + w, \\ d(i, j-1) + w, \\ d(i-1, j-1) + R(s[i], t[j]), \\ d(i-2, j-2) + F(s[i-1], s[i], t[j-1], t[j]) \end{cases}$$

Donde,

R(s[i], t[j]) = w_s es la longitud del camino más corto en el grafo G(V, A) del teclado para ir del carácter s[i] al carácter t[j] , si son diferentes y R(s[i], t[j]) = 0, si los caracteres s[i] y t[j] son iguales,

F(a,b,c,d)=w, si ab=dc, en otro caso no se tiene en cuenta (a, b, son caracteres de la cadena s y c, d caracteres de la cadena t)

Entonces EDUK(s, t) = d(n, m).

En la definición dada, se ha tomado un único costo para las operaciones de inserción, eliminación y transposición (w), lo que permite mantener la propiedad de simetría de la función definida y su condición de métrica, lo cual según [69, 70] es conveniente para el proceso de agrupamiento en que se utilizará.

Si se desea que el costo de la operación de sustitución juegue un papel de "separación" mayor, debe darse al costo de las operaciones de inserción, eliminación y transposición un valor mayor que uno. Por ejemplo, si este costo fuera dos, entonces EDUK(perez, pereo) = 4, lo cual hace que haya una mayor separación entre estas cadenas, que es lo que se desea.

La complejidad temporal para calcular esta distancia es un O(m.n), pues se guarda el grafo como un arreglo con las distancias de los caminos mínimos (la simetría de la matriz

de camino mínimos lo permite), por lo tanto, el cálculo de w_s se hace en tiempo constante (la búsqueda de los dos caracteres en un arreglo ordenado de longitud fija y luego, con los índices encontrados, el acceso a un elemento de un arreglo que representa el grafo).

Así, una vez definida esta distancia, se podrá, utilizando la técnica de agrupamientos, formar grupos con las cadenas de textos de manera que se pueda llevar a cabo la estandarización de las mismas.

Una primera prueba que se realizó con la distancia definida, fue la siguiente: se creó un conjunto de 65 cadenas que constituyen variaciones tipográficas de 16 cadenas. En estas variaciones, del total de errores el 10,9% son inserciones, el 12,7% son eliminaciones, el 7,5% lo constituyen transposiciones y el 69,9% son reemplazos. Estos porcentajes se corresponden con lo observado en la tabla 1.2. En el Anexo F se muestra la relación completa de las variaciones efectuadas a las 16 cadenas. Este conjunto de cadenas se desordenó, se le aplicó la técnica de agrupamiento descrita en el paso 3a del marco de trabajo (método PAM) con seis distancias: la distancia de Levenshtein-Damerau con costos para todas las operaciones de uno y dos, la distancia EDUK también con costos de uno y dos para las operaciones de inserción, eliminación y transposición, la distancia Q-Gram (bigram) y la distancia de Jaro.

Como resultado se puede observar las bondades de la distancia EDUK; en los dos casos se logra un agrupamiento más efectivo para la estandarización. Con la distancia EDUK se logra separar correctamente los grupos asociados a “rima” y “rema”. Además en el caso de “integras”, cuando el costo es dos va al grupo que le corresponde (el de “integrar”), pues la “s” está más cerca de “r” que de “l”.

En el propio Anexo F se muestra el resultado de las pruebas estadísticas de análisis de frecuencia y de la prueba no paramétrica de McNemar.

Se consideró cada grupo como una unidad para el procesamiento estadístico, así a los grupos que coincidieron con los originales, se les asignó el valor uno. A los que tuvieron algún error, se les asignó el valor cero.

El análisis de frecuencia muestra que con la distancia EDUK(2) se obtiene la menor cantidad de grupos mal formados y con la de Jaro se obtienen los peores resultados.

Se utilizó la prueba no paramétrica de McNemar en su versión exacta para comparar los resultados de las distancias contra el "patrón". Se crearon las tablas de contingencias, (véase Anexo F) y al analizar los valores de la significación de la comparación del patrón con las distancias EDUK (.5000 y 1.00) puede apreciarse la no existencia de diferencias significativas para un α del 5%. Luego el comportamiento de esas distancias es muy bueno, no tiene diferencias significativas con el patrón.

Al analizar los valores correspondientes a la comparación del patrón con las otras distancias puede apreciarse la presencia de diferencias significativas (Jaro), medianamente significativas (LD1 y LD2) y no significativas con QGram para un α del 5%. En este caso se puede señalar que la distancia EDUK se comporta de forma similar a Q-Gram.

En ocasiones las cadenas están compuestas por más de una palabra, por ejemplo, los valores del atributo "causa de baja" de la base de datos de la ONAT. Es frecuente también, que cadenas similares se escriban poniendo en diferentes posiciones las palabras que la componen. Siguiendo con el ejemplo del campo "causas de baja" se pueden observar las siguientes cadenas:

“titular con certificado médico” y “certificado médico del titular”

“chapistería y pintura” y “chapistería y pintura” (en la primera dos espacios entre chapistería e y)

“titular enfermo” y “enfermedad del titular”

“titular suspendido” y “suspendido el titular”

Como se puede apreciar, cada par de cadenas identifican la misma causa. Si se aplica la distancia EDUK a cada par de cadenas con costos uno y dos en las operaciones de inserción, eliminación y transposición, se obtienen los resultados que se muestran en la tabla 2.3.

Tabla 2.3 Distancia EDUK entre cadenas con varias palabras

s	T	$d_{EDUK}(s,t)$ (con w=1)	$d_{EDUK}(s,t)$ (con w=2)
“titular con certificado médico”	“certificado médico del titular”	24	47
“chapistería y pintura”	“chapistería y pintura”	1	2
“titular enfermo”	“enfermedad del titular”	23	35
“titular suspendido”	“suspendido el titular”	19	34

Se puede observar que cadenas que deben estar muy cerca, porque identifican el mismo objeto, se alejan por tener las palabras que la componen en un orden diferente.

Para tratar tales cadenas, se propone la siguiente idea: dividir cada cadena en las palabras que la componen (en lo adelante se utilizará el término token para generalizar), buscar la distancia EDUK de cada token de una cadena con los tokens de la otra y tomar la distancia mínima, estas distancias se suman y el resultado constituye la distancia entre las cadenas.

Formalizando esta idea:

Sean s y t dos cadenas y S = {s_1, s_2, … s_p}, T={t_1, t_2, … t_q} los conjuntos de los tokens que las forman, entonces:

$$d_{TOK}(s,t) = \begin{cases} 0 \quad si\ s = t \\ d_{TOK}(t,s) \quad si\ q < p \\ 1 + \sum_{i=1}^{p} m_i \quad en\ otro\ caso \end{cases}$$

Siendo $m_i = \min(d_{EDUK}(s_i, t_j)),\ j = 1..q$

Al mínimo de las distancias entre tokens se está sumando un uno que garantiza que solo cadenas iguales tengan distancia cero. Se ha tomado un valor entero para respetar la tradición de que la distancia de edición siempre es un valor entero.

La función d_{TOK}, definida de esta manera, constituye una distancia. La función es cero solo para el caso de cadenas iguales. Cuando se tengan dos cadenas con los mismos tokens en órdenes diferentes, la distancia entre ellos es uno. La propiedad de la simetría se garantiza por la parte recursiva de la definición.

La métrica EDUK es un caso particular de d_{TOK}, cuando las cadenas están formadas por un único token.

Si se busca la d_{TOK} para las cadenas que se muestran en la tabla 2.3 se obtienen los resultados que se muestran en la tabla 2.4.

Tabla 2.4: Distancia EDUK entre cadenas con varias palabras

s	t	d_{TOK}(s,t) (con w=1)	d_{TOK}(s,t) (con w=2)
"titular con certificado médico"	"certificado médico del titular"	5	7
"chapistería y pintura"	"chapistería y pintura"	1	1
"titular enfermo"	"enfermedad del titular"	12	21
"titular suspendido"	"suspendido el titular"	8	13

En este caso se usaron como símbolos separadores de tokens el espacio, el punto y la coma. Como se puede apreciar los valores son menores que los obtenidos anteriormente, lo que da una idea de acercamiento de estas cadenas, por lo que es más probable que al formarse los grupos, coincidan en el mismo.

2.2.2. Estandarización de las direcciones postales.

Las direcciones postales cubanas, a diferencia de las de otros países, son muy diversas en su estructura, por eso el intento de estandarizarlas es un proceso complejo. Aquí la estandarización se puede ver como el proceso de segmentar la dirección en sus partes componentes.

En [28] se plantea un trabajo interesante para segmentar direcciones postales, utilizando los modelos ocultos de Markov y se muestran resultados prometedores en comparación con otras técnicas de segmentación. El empleo de técnicas de aprendizaje automático se reportan en [62] para solucionar este problema.

En [13] se aplica también los modelos ocultos de Markov para la segmentación de la direcciones postales de la India, que se describen con una complejidad similar a las cubanas; en este caso se utilizan dos modelos, uno externo y otro interno. Para cada

elemento en que es posible segmentar, se construye un modelo interno que representa la estructura del elemento en particular. En este trabajo se adopta este mismo proceder.

Se hizo un estudio de direcciones reales cubanas que son almacenadas en la base de datos de los clientes de ETECSA. Se llegó a la conclusión de que es necesario separar las direcciones urbanas de las rurales, pues sus estructuras son totalmente diferentes. Se decidió abordar primeramente las direcciones postales urbanas y en trabajos futuros se abordarán las direcciones rurales.

A partir del estudio realizado, se determinó que en las direcciones urbanas pueden aparecer hasta 15 elementos diferentes:

Calle.	Piso.
Km.	Escalera.
Casa.	Apartamento.
Entre calle 1.	Reparto.
Entre calle 2.	Municipio.
Esquina.	Zona postal.
Edificio.	Código postal.
	Provincia

Estos elementos constituyen los estados del modelo oculto de Markov, y conforman el modelo externo. En el Anexo B se muestra un grafo simplificado de las relaciones que tienen lugar entre los estados y en el Anexo C se muestran todas las conexiones de cada uno de los elementos.

Se realizó un estudio de cada elemento para determinar los diferentes componentes que pudieran estar presentes y se determinó un modelo interno para cada elemento, (véase Anexo D).

Estos modelos internos difieren de los modelos propuestos en [13], pues en dicho trabajo se consideran los modelos internos con caminos paralelos y hay tantos caminos como elementos puedan aparecer en dicho componente; de tal manera que se tiene un camino con un elemento, otro con dos elementos y así sucesivamente (ver figura 2.2).

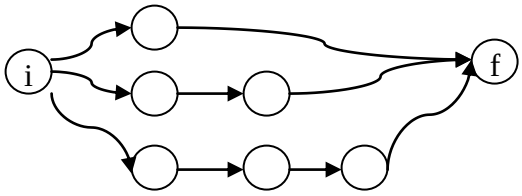


Figura 2.2

Sin embargo en el presente trabajo, los modelos internos pueden tener la misma complejidad que el modelo externo; tal y como se muestra en el Anexo D.

En [13] se señala que en la fase de entrenamiento se crean diccionarios para cada uno de los estados de los modelos, tanto para los internos como para el externo. En estos diccionarios se almacenan los símbolos que se pueden emitir en ese estado.

En el presente trabajo se crean estos diccionarios, utilizando la distancia d_{TOK}, de tal manera que una cadena se inserta en el diccionario, si no existe ninguna que tenga una distancia menor o igual a dos con la que se quiere insertar. El valor dos se ha tomado de manera empírica, indicando la posibilidad de permitir dos errores como máximo, para considerar cadenas similares. Se han obtenido resultados satisfactorios con el uso de este parámetro con valor dos; sin embargo, deben ser investigadas variaciones del mismo en

futuros trabajos. Esta misma técnica se utiliza cuando se busca en el diccionario, se considera la búsqueda exitosa cuando se encuentra una cadena con distancia menor que dos con respecto a la que se está buscando.

El método a seguir para la segmentación es, esencialmente, el propuesto en [13] :

Elegir las direcciones que sirven para el entrenamiento, buscando las más representativas.

Realizar el entrenamiento del modelo, en este caso se calculan las matrices denominadas A (matriz de probabilidad de transición de un estado a otro) y B (matriz de probabilidad de emisión de un símbolo en un estado determinado). Además se van construyendo los diccionarios de cada estado, con la particularidad de incorporar cadenas semejantes y no exactamente iguales.

Después de tener el modelo entrenado es conveniente tomar un grupo de direcciones para probarlo. Si el porcentaje de éxito es alto, se puede utilizar el modelo en la segmentación. Si no lo es, será necesario buscar nuevas direcciones para realizar un nuevo entrenamiento.

Es importante tener presente que no se puede considerar el modelo entrenado como permanente, pues para localidades diferentes habrá que hacer el proceso de entrenamiento a fin de actualizar los diccionarios.

Además, es necesario recordar que en el marco de trabajo propuesto, descrito en el epígrafe anterior, se plantea como paso previo a la estandarización de direcciones postales, la sustitución de determinados apócopes, abreviaturas, símbolos, etc., lo que puede reducir considerablemente el tamaño de los diccionarios y hacer la búsqueda mucho más eficiente.

En el capítulo 3 se dan detalles de la herramienta implementada para la realización de la segmentación de direcciones postales.

2.3. Solución propuesta para el tratamiento de los valores ausentes

Como se explicó en el capítulo 1, una de las tareas de la limpieza de datos es el reemplazo de valores ausentes, sobre todo si se trata de la carga de un almacén, pues tales valores pueden influir de forma negativa en los análisis que se hagan sobre los datos.

El reemplazo de valores ausentes es, en general, un proceso complejo y controvertido, pues se va a sustituir un valor desconocido, por uno que se determine, por algún método, pero que no es el valor real. Incluso pudiera pasar que el ausente fuera insustituible (recordar el ejemplo de la salsa del epígrafe 1.1.3), pero al usar la misma marca para valores ausentes y para nulos, ya no hay posibilidad real de saber con qué tipo de valor se está trabajando. Sólo en el caso de que se haya previsto tal diferencia por los analistas del sistema y que los operadores que entren datos estén conscientes de las mismas, se podrá diferenciar un ausente de un nulo. Para simplificar, en las explicaciones que siguen se denominarán valores ausentes como se ha hecho a lo largo del trabajo.

La mayoría de las técnicas de reemplazo de valores ausentes son técnicas estadísticas, lo que supone que los especialistas encargados de la tarea de la limpieza de datos, además de conocer los datos con que trabajan, también tengan conocimientos de esta disciplina.

Como existen múltiples técnicas, se propone a continuación un procedimiento que ayuda al reemplazo de valores ausentes, combinándolas y apoyado por un software que se describirá en el capítulo 3 del presente trabajo. El procedimiento no es completamente automático, pues el criterio del especialista que hace el análisis, no se puede ignorar y, en

muchas situaciones, la tarea a realizar dependerá del conocimiento que tenga el mismo sobre los datos.

Procedimiento para el reemplazo de valores ausentes

Paso 1: Determinar la base de datos y la tabla con la cual se trabajará.

Este paso es obvio, pero es necesario tener presente que la unidad con que se trabajará en el reemplazo de ausentes es una tabla.

Paso 2: Creación de una tabla auxiliar donde se hará el reemplazo de los valores ausentes.

Si se sustituyen los valores ausentes en la tabla original, es muy difícil reconocerlos, y revertir el proceso, sería casi imposible. Por lo tanto es necesario crear una tabla auxiliar. En el caso de que la sustitución de los valores ausentes se haga en la carga de un almacén de datos, la tabla auxiliar estaría en el área destinada a este proceso (staging area).

Paso 3: Determinación de algunas estadísticas necesarias

Se determinará en este paso la cantidad de ausentes en cada uno de los atributos de la tabla (y el porcentaje que representa del total de tuplas), el porcentaje de ausentes totales de la tabla y los patrones de valores ausentes.

Para estas estadísticas se tendrán en cuenta, en primer lugar, los valores NULL presentes en los diferentes campos. También se pueden tener en consideración los valores 0 (para los campos numéricos) y las cadenas vacías (para campos tipo cadena), pues, como se ha referido en el presente capítulo, es muy común en nuestros entornos de datos no utilizar la marca NULL para indicar ausencia de valores y sí emplear el cero o la cadena vacía.

Paso 4: Ignorar valores ausentes.

La variante más sencilla de eliminar los valores ausentes, es eliminar las tuplas que los contienen, esta técnica es muchas veces denominada “Análisis de Casos Completos” [100]. Esto es muy sencillo, pero solo aplicable cuando la cantidad de valores ausentes es poca, por ello se propone cuando los valores ausentes no sobrepasan el 1% del total [4, 82, 122, 123], aunque hay autores que plantean que se puede utilizar hasta el 5% [46].

Hay otro caso a tener en cuenta y es ignorar no todas las tuplas que tengan valores ausentes, sino aquellas que tengan mayor cantidad de ellos y que se consideren no utilizables. Para ello se define el concepto de umbral de uso (UU) que se fija en 0,5 (considerando que si más de la mitad de los atributos en una tupla están ausentes, la información que ella proporciona es de escaso valor) y se calcula para cada patrón de valores ausentes encontrado, el grado de incompletitud:

$GI = \frac{m}{n}$, donde n es el grado de la tabla (cantidad de atributos) y m la cantidad de valores ausentes en el patrón.

Se seleccionan los patrones cuyo grado de incompletitud sea mayor que el umbral de uso definido (GI > UU), se determinan cuántas tuplas cumplen con estos patrones y se propone su eliminación.

El umbral puede ser modificado en dependencia del conocimiento que el analista tenga de los datos.

Paso 5: Imputar valores ausentes a partir de valores conocidos.

En la literatura se reportan diferentes procedimientos estadísticos para imputar los valores ausentes, pero en ocasiones el valor ausente puede ser reemplazado por algún valor presente en la tabla que se analiza o en alguna otra que exista en la base de datos, o por el resultado de una expresión que se calcule con estos valores.

En este paso se solicita al analista que determine si tales valores o expresiones existen y, en caso de ser valores de otras tablas, generalmente hay que realizar un acople entre ambas tablas, lo que también se solicita en este paso.

Un ejemplo de este caso fue encontrado en la base de datos de la ONAT: Un dato que se tiene en la tabla de contribuyentes, es la fecha de alta; en ocasiones este dato estaba ausente, pero existe un atributo obligatorio en la tabla de pagos donde se refleja la fecha en que se efectúan los pagos. Se pudo sustituir las fechas de altas ausentes por la fechas del primer pago del contribuyente dado. Para ello fue necesario el acople entre las dos tablas, usando el código del contribuyente que está presente en ambas tablas, en la de pago como llave extranjera.

Paso 6: Reemplazo utilizando estimadores para atributos individuales.

En este paso se sustituyen los valores ausentes de un atributo por valores calculados a partir de los datos presentes. Se recomienda, cuando el porcentaje de valores ausentes no sobrepasa el 5% del total. Para aplicarlo, es necesario suponer que los valores ausentes se distribuyen de la misma forma que los valores presentes en los datos.

Los valores que se utilizan para sustituir los valores ausentes son:

Media: Se calcula a partir de los datos presentes y se sustituyen todos los datos ausentes por este valor. Esto hace que la media de los datos se mantenga. Se utiliza en el caso de valores numéricos.

Moda: Se calcula a partir de los datos presentes y se sustituyen todos los datos ausentes por este valor [100, 112]. Esto hace que la moda del conjunto de datos se conserve. Se utiliza en el caso de valores numéricos y categóricos. En caso de no existir la moda, esta técnica no es aplicable.

Mediana: Se calcula a partir de los datos presentes y se sustituyen todos los datos ausentes por este valor [100, 112]. Esto hace que la mediana se preserve. Se utiliza para el caso de valores numéricos.

Desviación Standard. Se trata de preservar la desviación estándar de los datos [112] y se calcula el valor que la mantenga a partir de los datos presentes, usando la siguiente expresión:

$$X_{n+1} = \frac{\frac{2*\sum_{i=1}^{n} x_i}{(n+1)^2} \pm \sqrt{\frac{4*\left(\sum_{i=1}^{n} x_i\right)^2}{(n+1)^4} - \frac{4*n}{(n+1)^2}\left(\frac{\sum_{i=1}^{n} x_i^{\,2}}{(n+1)} - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{(n+1)^2} - S^2\right)}}{\frac{2*n}{(n+1)^2}}$$

El valor calculado sustituye el primer valor ausente. Luego se calcula el siguiente valor teniendo en cuenta el ya calculado y se sustituye el segundo valor ausente, y así sucesivamente hasta que todos los valores estén reemplazados.

Este proceso hace que la varianza del conjunto de datos se mantenga y se utiliza para datos numéricos reales.

Método probabilístico basado en la distribución de los valores válidos: Se calcula el porcentaje de aparición de cada uno de los valores válidos, se utiliza el método para valores discretos (ya sean numéricos o alfanuméricos) y luego se reemplaza de manera aleatoria el mismo porcentaje de valores ausentes. El proceso de reemplazo se comienza por los valores con mayor porcentaje de aparición. Este método no se ha encontrado descrito de esta manera en la literatura revisada. El método contribuye a mantener la probabilidad de aparición de los valores válidos. Se puede aplicar en caso de que la moda falle o que la frecuencia de los valores no sea desproporcionada para alguno de ellos.

En todos estos casos se calculan la media, la mediana, la moda, la varianza, la desviación típica para los valores presentes y luego se calcula de nuevo, incluyendo los valores imputados. Se muestran estos resultados para que sirvan de base para tomar alguna decisión. En caso de confirmación se fijan los valores imputados en la tabla auxiliar.

Cualquiera de estos métodos tiene que ser aplicado con cuidado ya que puede introducir un sesgo en los datos, pues imputa un mismo valor para cada ausencia de una misma variable (es decir, que si una variable está ausente en 20 registros diferentes, habrá 20 nuevas ocurrencias de la “media” en esa variable), lo cual puede afectar a otras técnicas de minería de datos hasta el punto de invalidarlas (por ejemplo, si se aplicara una “clasificación”, la súbita aparición de más

ocurrencias de los valores medios, podría causar mucha “afluencia” de registros hacia una clase en particular).

Paso 7: Reemplazo de valores ausentes usando relaciones entre variables.

Como se vio en el capítulo 1, se reporta reiterativamente en la literatura el uso de la regresión como un método para imputar valores ausentes, por el hecho de que tiene en cuenta no solamente los valores de una variable en particular, sino la interrelación que puede existir entre las diferentes variables.

A partir del patrón de valores ausentes se determina si es monótono o no. En caso de ser monótono, se propone el uso de la regresión como técnica su reemplazo.

Si el valor que ha de imputarse es numérico, se puede emplear la regresión múltiple. En caso de que sea una variable categórica, podría emplearse la regresión logística y hacer el reemplazo según la probabilidad que el modelo de regresión estimado otorgue a cada categoría para el sujeto en cuestión [17].

Si los patrones de valores ausentes no son monótonos, se propone usar el método MCMC (Markov Chain Monte Carlos) [127] para imputar algunos valores y convertir los patrones en monótonos para luego aplicar los métodos de regresión.

Paso 8: Aplicación de otras técnicas para el reemplazo de valores ausentes.

Existen otras técnicas que pueden ser aplicadas:

Imputación Hot Dock [4]: Los datos ausentes son reemplazados con valores seleccionados aleatoriamente, presentados en un grupo de datos completos similares; o sea, identifica los casos más semejantes al caso del valor perdido y

sustituye el valor perdido por ese [6]. Es una técnica semejante a la aplicada en los Sistemas Basados en Casos, pero aquí es necesario utilizar alguna métrica para determinar cuándo los casos son similares.

Árboles de Decisión [141]**:** Los árboles de decisión sustituyen los valores ausentes utilizando algoritmos tales como ID3 [113] o C4.5 [114]. Se construye un clasificador con las tuplas que no tengan valores ausentes y se toma como atributo clase el que tiene valores ausentes.

El procedimiento descrito se podrá enriquecer con nuevas técnicas de imputación, como la descrita en [129] que utiliza reglas de asociación ordinales para este propósito.

A partir de este procedimiento se creó un software para el reemplazo de valores ausentes. En dicha herramienta están implementadas todas las técnicas hasta la regresión.

2.4. Conclusiones parciales

Desde el punto de vista teórico se han obtenido los siguientes resultados:

La distancia EDUK y su generalización d_{TOK}, que favorecen al problema de estandarizar cadenas de caracteres con errores tipográficos. Los resultados de las pruebas realizadas demuestran la efectividad de estas distancias.

El marco de trabajo para la estandarización de cadenas, que posee valor metodológico y sirve de base para el desarrollo de una herramienta que permita realizar esta tarea de la limpieza de datos.

La formalización del procedimiento para la sustitución de valores ausentes utilizando la probabilidad de los valores válidos y se propone un procedimiento general para el tratamiento de los valores ausentes en las bases de datos, que agrupa las principales

técnicas para realizar el reemplazo de valores ausentes, guiando a los especialistas en su proceder; lo que le atribuye valor metodológico. Este procedimiento fundamenta el diseño e implementación de una herramienta para el reemplazo de valores ausentes.

La creación de la herramienta de detección de errores DBAnalyzer, que permitió establecer la taxonomía de errores de las bases de datos cubanas.

3. APLICACIONES Y EXPERIMENTACIÓN

En este capítulo se describen las características generales de las herramientas obtenidas para la limpieza de datos, y se muestran los resultados de su aplicación a diferentes bases de datos.

3.1. Concepción de las herramientas y sus características generales

Las herramientas DBAnalyzer, DBStandard, y DBNulos han sido confeccionadas usando el lenguaje de programación Object Pascal.

Estas herramientas permiten la conexión con diferentes bases de datos, para esto se utiliza el componente ADO que brinda el Delphi y en todas ellas, esto es lo primero que se realiza. Poseen un ambiente interactivo en que es importante la participación del administrador de la base de datos para desarrollar el paso de la limpieza que se esté llevando a cabo.

En una etapa posterior del trabajo se ha utilizado el software libre para desarrollar las herramientas de limpieza y se cuenta con EDPOS, que es un software realizado en Java para la segmentación de direcciones postales y con la segunda versión de DBNulos también en este lenguaje.

3.1.1. El DBAnalyzer

La herramienta DbAnalyzer está destinada a realizar el análisis de datos, integrando los dos enfoques propuestos en [115] : los perfiles de datos y la minería de datos.

El DbAnalyzer obtiene el perfil de los datos, calculando diferentes estadísticas que permiten revelar información importante sobre los datos, de acuerdo a su tipo, de la siguiente forma [83] :

Para todos los datos

- Tipo de dato, tamaño,
- Cantidad de valores ausentes,
- Unicidad,
- Por ciento de valores por defecto.

Para datos discretos

- Cardinalidad,
- Valores diferentes y su porcentaje.

Para datos numéricos

- Valores máximo y mínimo,
- Valor medio, varianza, moda, mediana, desviación estándar
- Contar ceros (pueden indicar valores ausentes).

Para datos tipo cadena

- Contar cadenas vacías (pueden indicar valores ausentes).

Además esta herramienta implementa la detección de errores, utilizando la técnica de reglas de asociación de la minería de datos, en particular se implementan dos algoritmos para el descubrimiento de reglas de asociación: el algoritmo que descubre reglas de asociación ordinales descrito en [97] y el clásico algoritmo de descubrimiento de reglas de asociación Apriori [1], y se utiliza el principio de que si una regla es descubierta, es porque la misma se cumple en un alto porcentaje de los registros de datos, por tanto debe

examinarse el porcentaje restante pues las excepciones pueden ser errores. También para completar los metadatos se incorporaron a la herramienta otras pruebas que son de utilidad en las bases de datos de nuestro país, por ejemplo: la validación de los números de carné de identidad, la validación de fechas dentro de determinados rangos y el chequeo de unicidad en la concatenación de varios atributos.

Para lograr su extensibilidad se diseñó en tres capas:

- Una capa de manejo de datos, que permite la conexión en la versión actual con bases de datos en cualquier gestor.
- Una capa para el análisis de datos, donde las clases se ocupan de encontrar los estadígrafos y las reglas de asociación especificados anteriormente.
- Una capa de interfaz con el usuario.

Para el desarrollo de la capa de análisis de datos, se creó un conjunto de clases que se muestra en el Anexo E.

Aplicaciones de la herramienta DBAnalyzer

La herramienta ha sido aplicada en varias bases de datos con el objetivo de lograr su validación, y demostrar su efectividad y utilidad para realizar la detección de errores.

La primera aplicación de la herramienta [99], se realizó sobre:

- Las bases de datos Zafra y ZafraNueva en SQL Server, proporcionadas por el MINAZ Provincial,
- la base de datos del Sistema de control de estudiantes de la UCLV en ACCESS y
- las bases de datos del Sistema de control de la mortalidad nacional de los años 1999 y 2003, confeccionadas en FoxPro.

La base de datos Zafra está relacionada con el sistema de contabilidad, referido a las diferentes instituciones involucradas en el proceso de la zafra. Esta base de datos tiene un total de 406 tablas y un tamaño en disco de 64 MB.

Como resumen de los principales problemas detectados con la herramienta en esta base de datos, se pueden señalar:

Registros que faltan: dado por el hecho de que existen 131 tablas vacías, 14 tablas con solo una tupla, dos tablas con dos tuplas y seis tablas con tres tuplas.

Campos que faltan: existen en la base de datos seis tablas con un campo vacío.

Información incorrecta entrada al sistema: cuatro tablas en que hay campos que representan descripciones en que la información es incorrecta.

Uso y significado inconsistente de diversos códigos: en cuatro tablas hay campos en que se hace un uso inconsistente de códigos.

Uso inconsistente de nulos, vacíos y espacios: en seis tablas son usados de forma inconsistente el valor 0 en el caso de campos numéricos y la cadena vacía en el caso de campo de tipo texto.

La base de datos ZafraNueva guarda la información referente al corte y alza de caña, así como de las estadísticas de los diferentes centrales, esta base de datos tiene un total de 169 tablas y un tamaño en disco de 664 MB. Su análisis brindó los siguientes resultados:

Registros que faltan: existen 18 tablas vacías.

Uso inconsistente de nulos, vacíos y espacios: nueve tablas presentan este problema, una en 8 de sus campos, una en 5 de sus campos, otras dos en 3 de sus campos y el resto en 1 campo de cada tabla.

El análisis de la base de datos de Control de estudiantes, tiene 19 tablas, los principales errores encontrados son:

Registros que faltan: tiene una tabla con un solo registro.

Carné de identidad incorrecto: siete de las tablas en que aparece este campo tienen errores, en total se detectaron en estas tablas 234 valores incorrectos de carné de identidad.

Registros duplicados: en una tabla aparecen dos registros duplicados.

Información errónea entrada al sistema: en dos tablas.

La base de datos del Control de mortalidad del año 1999 (defu99) consta de dos tablas:

Una tabla es un codificador de las causas de muerte y los errores se refieren a:

Uso y significado inconsistente de diversos códigos: este problema está presente en cinco registros de dicha tabla en que un mismo código posee varias descripciones diferentes.

Diferentes códigos con el mismo significado: se presenta este error en 35 registros en que el mismo valor para el campo descripción está asociado a diferentes códigos.

En la otra tabla que almacena las defunciones (79 499 registros) los errores se refieren a:

Campos que faltan: diez campos vacíos en las 79 499 tuplas, además de tres campos en que 13 876 registros son vacíos.

Uso inconsistente de nulos, vacíos y espacios: 11 campos cuyos datos tienen valor 0, indicando ausencia de información.

Registros duplicados: 20 registros en que se repite parte de la información, en este caso están duplicados los nombres y apellidos del fallecido.

La base de datos de Control de mortalidad del año 2003 (defu03), consta de 2 tablas, una es el codificador de las causas de muerte y se presentan los mismos tipos de errores que en la anterior.

En la otra tabla (78 434 registros) los errores son:

Información errónea entrada al sistema: existen 2 campos "apellido1" y "apellido2" en que se usan abreviaturas para los apellidos; en ocasiones, varias abreviaturas son utilizadas para un mismo apellido; además también hay varias formas en que aparece un mismo apellido, a causa de errores al teclearlo.

En el campo "apellido1" se contabilizan 633 usos de abreviaturas por diferentes apellidos y en algunos casos son varias las abreviaturas que se utilizan para un mismo apellido, por ejemplo, por Rodríguez, Rdguez y Rguez. Además existen en este campo 84 apellidos con errores tipográficos.

En el campo "apellido2" hay 633 registros en que usan abreviaturas y se incurre también en el mismo problema de que se utilizan diferentes formas de abreviaturas por el mismo apellido. Hay 69 valores con errores tipográficos.

Carné de Identidad incorrecto: en el campo "noident" existen 52 números incorrectos de acuerdo al chequeo de los seis primeros dígitos, que representan la fecha de nacimiento del paciente.

Campos que faltan: en ocho campos hay valores ausentes en un promedio de 11699 artículos.

Uso inconsistente de nulos, vacíos y espacios: en dos campos se utiliza el cero para indicar información ausente.

El software también fue aplicado en la base de datos correspondiente a una sucursal del Banco Provincial de Ahorro en Villa Clara, los resultados están expuestos en [92]. Se analizó la tabla que recoge la información de los clientes en 34 campos y que cuenta con 20 724 artículos y la tabla "Centro de pagos" con 21 campos y 46 tuplas.

Los errores más representativos son:

Registros duplicados: la cardinalidad del campo "carné de identidad" de la tabla "Clientes" es de 18 878, lo que indica que hay números de carné repetidos, por ser este campo único. Lo mismo ocurre con el campo "Número de serie" y con el campo "Identificador del centro de pago" de la tabla "Centro de pagos".

Carné de identidad incorrecto: 45 números de identidad captados con longitud menor que 11, un total de 148 representados de forma incorrecta.

Campos que faltan: en el campo "Carné de identidad" y en el campo "Número de serie" hay 1817 cadenas vacías, en el campo "Calle" se detectaron 1826 valores ausentes y en el campo "Sexo" se detectaron 1832.

Información errónea entrada al sistema: además de los detectados en el campo "Carné de identidad", también se detectaron en el análisis realizado 212 números de serie incorrectos y repetidos, 1166 nombres de calles captados de forma incorrecta y descripciones distintas, asociadas a una misma sucursal.

Uso y significado inconsistente de diversos códigos: aparecen dos códigos asociados a la misma descripción de sucursal.

Otra de las pruebas se realizó con la base de datos de Recursos Humanos de la Universidad Central de Las Villas [90], en la que se probó el análisis a partir de reglas de asociación y donde se encontraron casos interesantes como:

Se descubrió la regla de asociación ordinal en el análisis de la tabla "RH_AjustesSubmayorRetenciones",

Valor_Deduccion < Saldo (s = 1 y c =0.99)

El significado de esta regla indica que la deducción que se hace en un salario, tiene que ser menor que el salario total; esta debe ser una regla del negocio, pero no está implementada en la tabla y aparentemente tampoco en el sistema, porque se detectó que dos registros (69 y 196) la incumplen.

En esta misma tabla fueron descubiertas otras reglas, por ejemplo:

Saldo > Ajuste (s = 1 y c =0.96)

Incumpliéndose en los registros 69, 105, 110, 116, 196, 232, 237, 243 los que al ser analizados por el especialista en el negocio resultaron realmente erróneos.

En la misma base de datos pero en la tabla "RH_Detalles_Reporte_Nominillas_Mov" se descubrieron las siguientes reglas:

Importe > Salario_Acumulado (s = 1 y c = 0,99), que no se cumple en : 7924, 14453, 14454, 49760, 14455

Importe > Tarifa_Divisa (s = 1 y c = 0,99), que no se cumple en: 14453, 49760, 14455, 14454

Importe > Divisa_Factura (s = 1 y c = 0,99), que no se cumple en: 14453, 14454, 14455, 49760

En el análisis hecho por los especialistas del negocio, se corroboró que la primera y la tercera reglas eran reales y los registros señalados realmente erróneos. En el caso de la segunda regla, esta no fue avalada y los registros señalados no contenían valores incorrectos.

El DBAnalyzer muestra cuáles son los registros que incumplen las reglas encontradas, de forma que el especialista conocedor de los datos pueda revisar y determinar cuáles de estos datos constituyen verdaderamente un error y proceder a su corrección.

Combinar en esta herramienta DBAnalyzer los dos procedimientos: el análisis de atributos individuales por su perfil y el análisis de pares de atributos por las reglas de asociación, permite al especialista tener en cuenta que los atributos que coincidan como errores potenciales en ambos casos son los más propensos a ser verdaderos errores en los datos analizados.

También se realizó la detección de errores en la base de datos de Multas MulMaster, la base de datos Integral de la UCLV y la base de datos del Sistema de control de estipendio de la UCLV.

3.1.2. El DBStandard

Esta herramienta realiza la limpieza de datos de los campos de tipo cadena de caracteres, implementando los diferentes pasos del marco de trabajo expuesto en el epígrafe 2.2 del capítulo II. Igual que la herramienta DBAnalyzer esta se implementa en Borland Pascal y presenta una interfaz amigable que permite que, de forma interactiva, el administrador de los datos decida cómo realizar la limpieza de datos.

Se desarrolló otra versión de prueba de la misma que permitió realizar los experimentos para facilitar la validación de los resultados de emplear la distancia d_{TOK}, calculando el coeficiente de silueta de los grupos formados y los coeficientes de precision y recall para determinar F-Score, como medida externa que ofrece un criterio sobre la bondad de los grupos formados.

3.1.2.1. Experimentos realizados

Para probar la efectividad del agrupamiento de la distancia d_{TOK}, se tomaron dos conjuntos de datos reales: “apellidos” de una base de datos de un censo sobre uso de equipos electrodomésticos realizado en Villa Clara (Cuba) y “causa de baja” de los contribuyentes de la ONAT del municipio de Ranchuelo (Villa Clara, Cuba). Para poder trabajar con precisión, se tomaron 100 de los apellidos donde estuvieran representados los tipos de errores en la proporción descrita en la tabla 1.2. Estas cadenas son relativamente cortas. En el caso de las causas se tomaron todas (125, sin repeticiones), aquí las cadenas tienen una longitud mayor que los apellidos y la naturaleza de los datos es diferente.

Se determinó por personal experto que en los apellidos había 42 categorías, o sea de los 100, solo había 42 apellidos diferentes y de las 125 causas, solo 39 eran diferentes.

Utilizando el método PAM, se hicieron varias pruebas de agrupamiento, variando el número de grupos (que en este método es un dato), desde 10 a 42 en el caso de los apellidos y desde 25 a 39 en el caso de las causas. Además se utilizaron tres variantes de los costos: una en la que los costos de inserción, eliminación y transposición era igual a uno, otra en que se le dio valor dos y una tercera en que los costos tienen valor tres. A estas distancias se les ha denominado D_{TOK-1}, D_{TOK-2} y D_{TOK-3} respectivamente.

Se midieron en cada agrupamiento dos parámetros: el coeficiente de silueta y el F-Score.

El coeficiente de silueta es una medida propuesta por Kauffman y Rousseeuw para indicar la pertenencia de un elemento al grupo [63]. Puede tomar valores en el intervalo [-1, 1], y mientras más se acerque a uno indica que los grupos son más cohesionados. Como se observa en las figuras 3.1 y 3.2, los valores de este coeficiente van aumentando en la medida que el número de grupo se va acercando a la cantidad de categorías reales

que hay en los datos. Esto hace que esta medida interna pueda utilizarse en la práctica para saber el número de grupos a considerar.

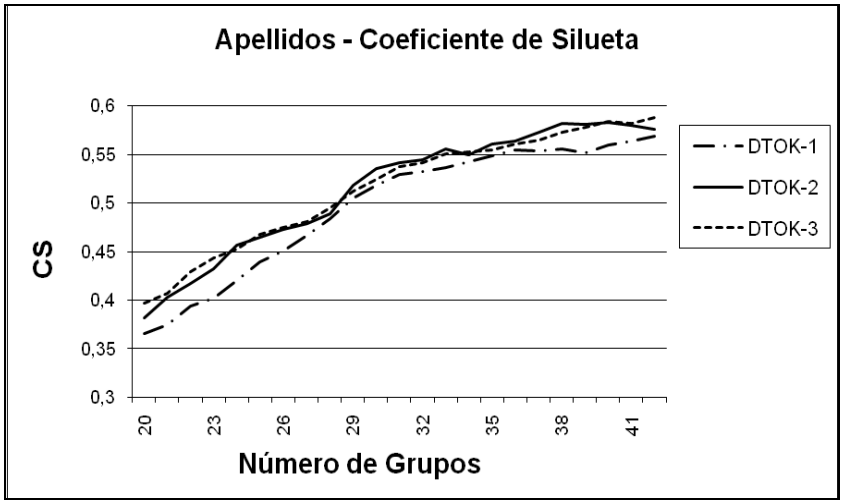


Fig. 3.1 Gráfico del coeficiente de silueta para el campo Apellido

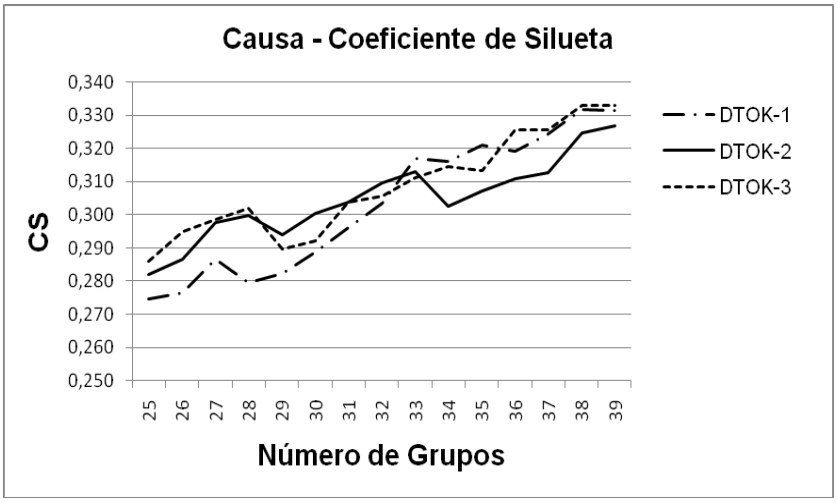


Fig. 3.2 Gráfico del coeficiente de silueta para el campo Causas

Este coeficiente da una idea de grupos compactos (mejor en el caso de los "apellidos" que las "causas"), pero no brinda información sobre la cantidad de cadenas que pueden ser estandarizadas a partir de este proceso. Por ello se decidió usar un coeficiente más relacionado con la recuperación de la información que con la formación de grupo: el F-Score (o f-measure). Tal índice se propone en algunos trabajos en la literatura para medir

efectividad en la recuperación de pares de cadenas semejantes [22, 71] y en el Anexo G se ha descrito la forma de calcularlo.

En la medida que F-Score aumente, indica una mejor recuperación. En las figuras 3.3 y 3.4 se observa cómo cambia este valor con la cantidad de grupos y que, en el caso de los "apellidos", aumenta en la medida en que la cantidad de grupos aumenta, lo que está en estrecha relación con el coeficiente de silueta, pues los grupos son más homogéneos, debido a que las longitudes de las cadenas no presentan una diferencia significativa.

Para las "causas" no se observa un incremento del F-Score, oscila alrededor de 0.45, esto se puede explicar porque la longitud de las cadenas es muy disímil (la más corta tiene longitud 5 y la más larga longitud 60), lo que hace que las distancias entre ellas tengan una gran variabilidad. Si se observan los grupos formados, se puede apreciar que las cadenas que constan de una sola palabra, se han agrupado en un único grupo [85].

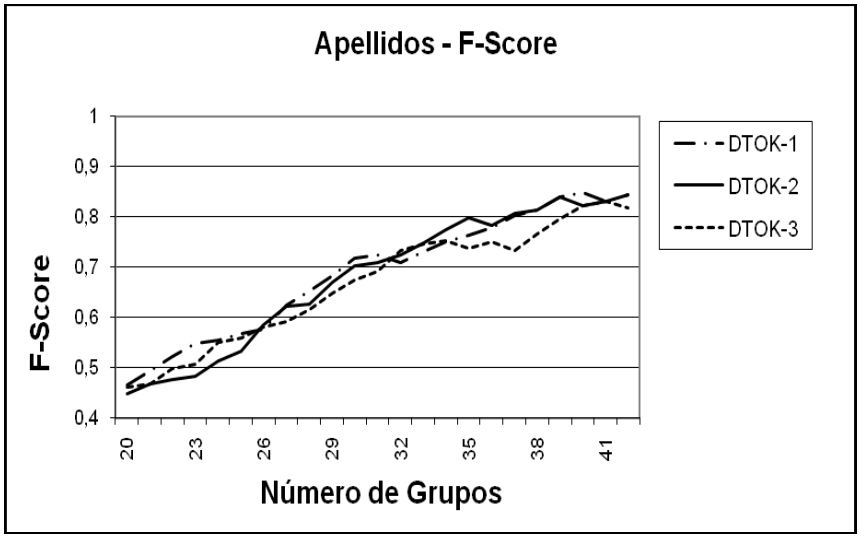


Fig. 3.3 Gráfico del índice F-Score para Apellido

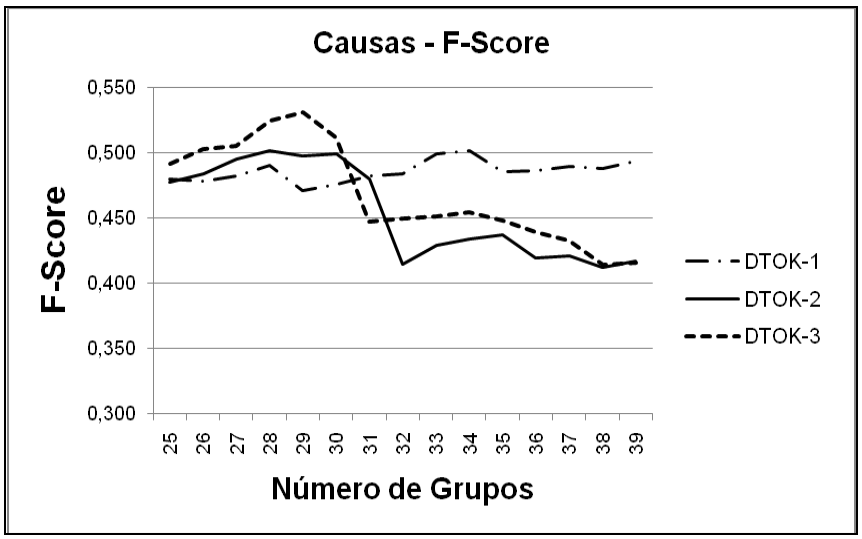


Fig.3.4 Gráfico del índice de F-Score para Causas

Aunque los valores obtenidos del coeficiente de silueta y de F-Score son considerados bondadosos, se compararon estos resultados con las otras distancias que se han mencionado. En las figuras 3.5 y 3.6 se observa una comparación del coeficiente de silueta con las distancias Q-Gram, Levenshtein y Jaro que son reportadas como exitosas en la determinación de cadenas semejantes. Como se aprecia, los valores obtenidos por la distancia d_{TOK} en 3.5 son comparables a los de las distancias Q-Gram, Levenshtein-Damerau y Jaro.

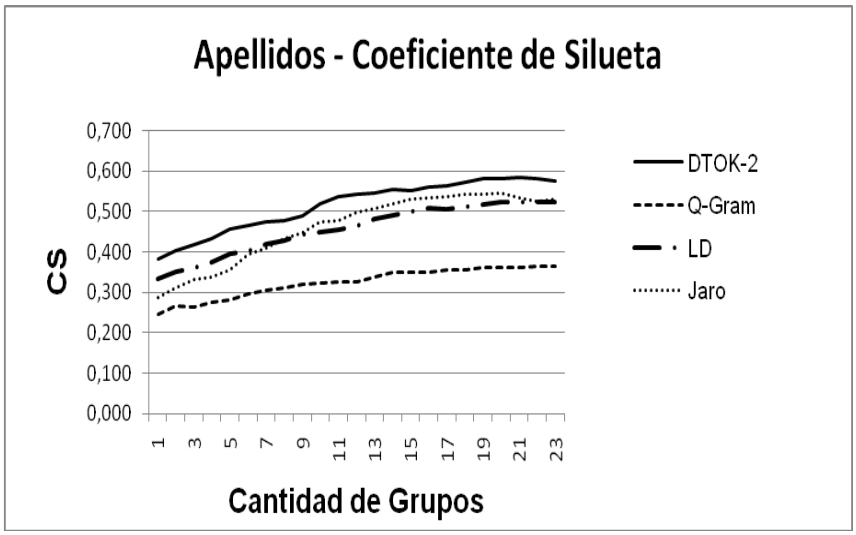


Fig. 3.5 Coeficiente de silueta del campo Apellido

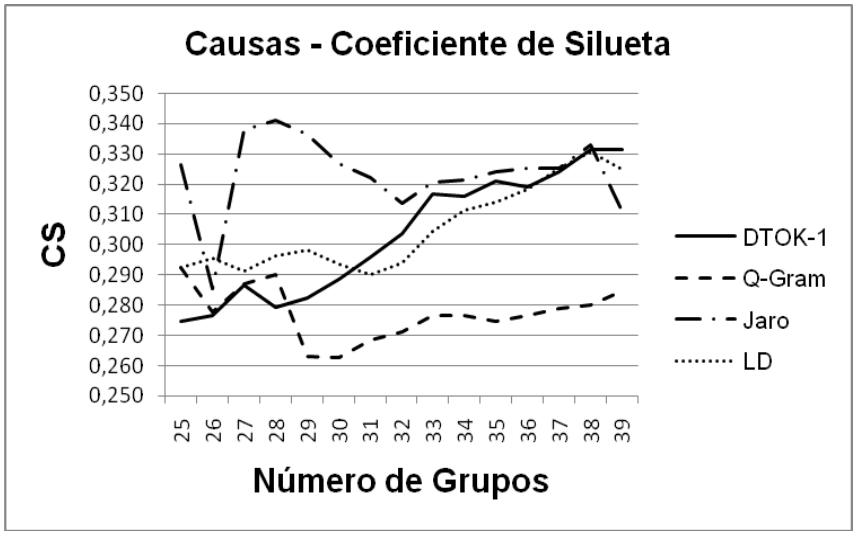


Fig. 3.6. Coeficiente de silueta para el campo Causas

En las figuras 3.7 y 3.8 se observa una comparación del F-Score, aquí se puede apreciar que la distancia propuesta tiene igual desempeño que las conocidas, es necesario destacar que para cadenas cortas la distancia d_{TOK} tiene un mejor comportamiento.

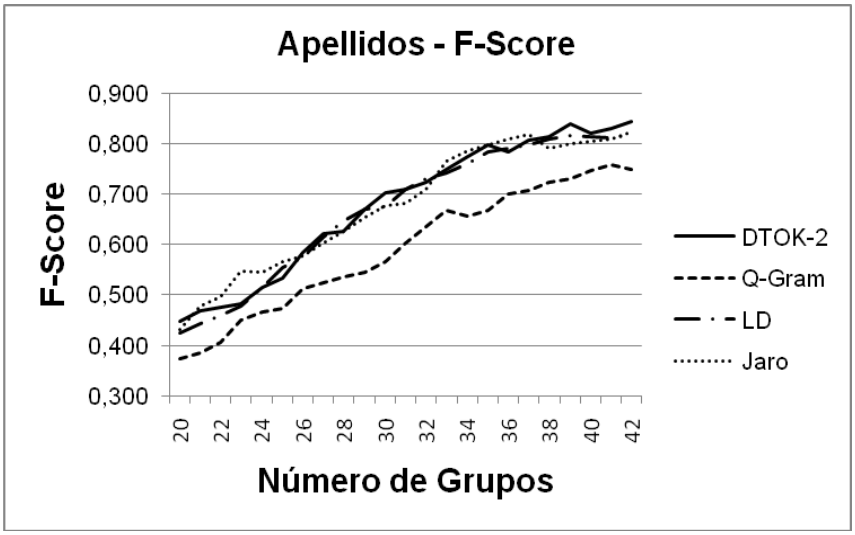


Fig. 3.7. Comparación del F-Score del campo "Apellido" con otras distancias

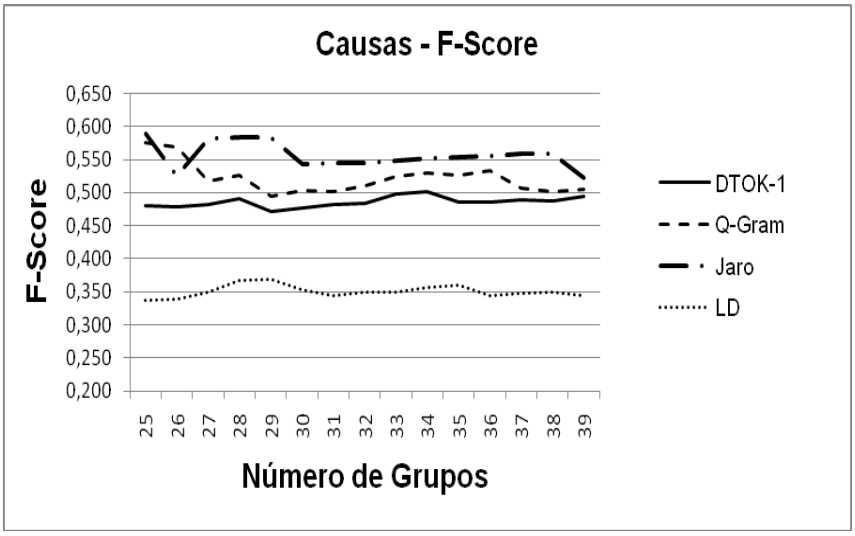


Fig. 3.8 Comparación del F-Score del campo "Causa" con otras distancias

Como conclusión de este experimento se mostró que la distancia d_{TOK} tiene un comportamiento comparable con otras distancias utilizadas para este mismo objetivo y en algunos casos, superior.

Una muestra de los resultados obtenidos en la estandarización para el campo Causa de la base de datos de la ONAT se exponen a continuación [85]:

Elementos del Grupo 4	
reparación del motor	0.447
reparación del motor	0.446
reparación del motor	0.380
reparación del vehículo	0.274
reparación del equipo	0.266
reparación del equipo y pintura	0.201
reparación del coche	0.188
titular suspendido por reparación del motor	0.082
reparación media	0.036
por suspensión del titular	0.153

Elementos del Grupo 5	
chapistería y pintura	0.808
chapistería y pintura	0.785
chapistería y pintura	0.774
chapistería	0.283

Elementos del Grupo 10	
Certificado del titular	0.403
certificado del titular	0.395
certificado médico del titular	0.384
certificado médico del titular	0.375
suspendido el titular	0.218

Elementos del Grupo 11	
problemas con el equipo	0.500
problemas con el equino	0.479
por presentar problemas con el equipo	0.364
problemas con el animal	0.106

Elementos del Grupo 12	
enfermedad del equino	0.511
enfermedad del equino	0.452
enfermedad del anima	0.419
enfermedad del caballo	0.344
por enfermedad del titular	0.146
enfermedad del titular	0.102
por presentar enfermedad el equino	0.058
muerte del equino	-0.027

Elementos del Grupo 21	
resolución 73 2005 art.36	0.815
resolución 73 05 art. 36	0.792
resolución 73 2005 arti.36	0.791
resolución 73 2005 art .36	0.790

resolución 73 2005 art.36	0.781
resolución 73 art.36	0.746
resolución 73art.36	0.720
por resolución 73 2005 art.36	0.707
resolución 73artic.36	0.672
resolución 23arti.36	0.619
resolución 86 2002	0.432

Como puede observarse por simple inspección, se nota que las cadenas han quedado agrupadas con una cierta lógica y pueden ser usados los grupos para realizar sustituciones que conlleven a la estandarización de las mismas. A la derecha de cada cadena se muestra el coeficiente de silueta, en la medida que este valor se acerque a uno indica que la cadena pertenece más fuertemente al grupo. En la muestra hay pocas cadenas con coeficientes de silueta pequeño o negativo, por ejemplo "muerte del equino" en el grupo 12, o la cadena "resolución 86 2002" en el grupo 21. Estos son los casos en que la decisión del analista de datos es importante para verificar si pertenece o no la cadena al grupo.

Con este procedimiento es muy fácil realizar el siguiente paso de sustitución de las cadenas para lograr la estandarización, por ejemplo, las cadenas del grupos 21 deben ser sustituidas por "resolución 73 del 2005, articulo 36", así la estandarización se realiza buscando en el grupo y no en la tabla completa de la base de datos.

Otra aplicación que resultó interesante es la realizada en el Departamento de Anatomía Patológica del Hospital "Arnaldo Milián Castro" de Villa Clara [89].

En la base de datos Necropsias, los patólogos escriben el resultado de sus informes de necropsias lo que provoca distorsiones al expresar los mismos resultados de una a otra. En dicha base de datos existe una tabla denominada "Protocolos", donde hay varios campos que tienen las características mencionadas, pero solamente se presentan algunos

resultados obtenidos con el campo CDM (Causa Directa de Muerte). Este campo tiene 57 cadenas diferentes.

Al ejecutar el procedimiento de estandarización con 10 grupos, se obtuvieron agrupamientos que realmente muestran la misma causa escrita de diferentes formas. Ejemplos de algunos grupos se muestran en la tabla 3.1.

Tabla 3.1 Ejemplos de grupos obtenidos con el DBStandard

1	shock mixto (septico e hipovolémico) shock mixto (séptico/hipovolémico) shock mixto(septico e hipovolemico)
2	arritmia cardiaca arritmia cardíaca arritmia cardiaca irreversible arritmia cardíaca irreversible arritmia irreversible arritmia ventricular irreversible
3	hipertension endocraneana hipertensión endocraneana hipertensión endocraneana(hemorragia protuberancial de duret) hipertension endocraneana. hemorragia de duret hipertensión endocraneana. hemorragia de duret del puente hipertensión endocraneana. hemorragia de duret izquierda. hipertensión endocraneana. muerte encefalica
4	disfunción encefálica (muerte encefálica) disfunción ventricular aguda disfuncion ventricular izquierda aguda disfunción ventricular izquierda aguda distres respiratorio del adulto insuficiencia cardio-respiratoria aguda insuficiencia cardio-respiratoria insuficiencia cardio-respiratoria aguda insuficiencia cardiorrespiratoria aguda insuficiencia respiratoria aguda insuficiencia ventricular aguda

Como se puede observar, se han agrupado causas de muerte que han sido escritas de formas distintas que hacen que en el análisis de los datos sean tomadas como causas de muerte diferentes. Después de realizar las sustituciones quedaron sólo 37 causas de muerte.

3.1.2.2. Estandarización de Direcciones

Se construyó una herramienta denominada EDPOS, la cual realiza la segmentación de direcciones postales, siguiendo el método explicado en el epígrafe 2.2.2. Algunas consideraciones de implementación se brindan a continuación.

La herramienta está programada en Java, comenzando así un proceso de migración de las herramientas realizadas a software libre.

Las sustituciones que se plantea realizar en las direcciones para lograr un primer nivel de estandarización, se hacen con ayuda de un fichero XML que consta de dos etiquetas fundamentales: “qué cambiar”, “por quién cambiar”. Este documento puede contener inicialmente algunos cambios comunes, establecidos por la experiencia de los analistas y puede ser enriquecido a lo largo del tiempo. Además, es importante que en determinados casos se pueda decidir no utilizar alguno de los cambios propuestos en el documento. La utilización de un documento XML viene dada por el hecho de no depender de un sistema operativo, ni de un gestor de datos específico, de tal manera que dicho documento sea fácilmente portable.

Los conjuntos de entrenamiento también se almacenan en ficheros XML. El nombre de las etiquetas coincide con el nombre de los estados de cada uno de los modelos. Es importante, en el momento de confeccionar estos ficheros, respetar el orden en que aparecen los elementos en cada una de las direcciones.

El diagrama de las clases fundamentales se muestra en el Anexo E.

En este diagrama se aprecia la existencia de las clases Nodo y Arista, como elementos constitutivos del HMM, además de la clase Diccionario que sirve para implementar los diferentes diccionarios con que trabaja el modelo.

La conexión con la base de datos se logra mediante el controlador JdbcODBC.

El software permite, además, almacenar el modelo entrenado de manera que pueda ser reutilizado para posteriores procesos de segmentación.

Esta herramienta se utilizó para la segmentación de direcciones postales de la base del Censo de efectos electrodomésticos de Villa Clara. Se escogieron Consejos Populares que fueran del sector urbano. Se utilizó un grupo de alrededor de 500 direcciones para entrenar y cuando se aplicó el modelo, se obtuvo una efectividad de un 96% de direcciones bien segmentadas.

3.1.3. El DBNulos

3.1.3.1. Concepción de la herramienta DBNulos

La herramienta DBNulos fue concebida inicialmente como una aplicación con las características generales que se describen en el epígrafe 3.1 y así se realizó el registro de propiedad. Sin embargo, al tratar de introducir su utilización, se observó que muchos de los especialistas que trabajan con las bases de datos, carecen de los conocimientos estadísticos que son necesarios para utilizar dicha herramienta. Se decidió realizar una nueva versión en forma de Asistente que vaya conduciéndolos, a través de su ejecución, en la selección adecuada del método para realizar el reemplazo de los valores ausentes.

De esta forma y teniendo en cuenta la migración ya comenzada en el método de estandarización de direcciones, se produce el Asistente DBNulos, que implementa los pasos descritos en el epígrafe 2.3, en el lenguaje Java.

La versión actual de la herramienta implementa los siete primeros pasos posibilitando realizar imputaciones de valores ausentes con los métodos de interpolación lineal y múltiple. No están implementadas la regresión logística y otras técnicas más avanzadas.

3.1.3.2. Experimentación

Para probar el método propuesto, se tomó la base de datos del repositorio de la UCI "Adults". Esta base de datos tiene 11 campos: age, fnlwgt, workclass, education, education-num, relationships, race, sex, hours-per-week, native-country, marital-status. De ellos son numéricos age, fnlwgt y hours-per-week y el resto de los campos son ordinales. "Adults" que contiene originalmente 32562 registros, se tomaron 4480 al azar, para trabajar con una muestra más manejable, de tal manera que en la base de datos no quedaran valores ausentes. A la base de datos se le añadió un campo autonumérico que constituyera la llave primaria de cada registro (el original no contiene ningún atributo que pueda ser tomado como llave primaria) para poder realizar los cálculos con precisión.

El experimento consistió en generar aleatoriamente un porcentaje de valores ausentes. Se hicieron 4 pruebas principales, generando el 5%, 10%, 15% y 20% de ausentes, para cada uno de los por cientos se hicieron tres pruebas. Luego con la herramienta descrita se completaron todos los datos, utilizando las diferentes variantes del procedimiento. Finalmente, se comprobó cuál era la exactitud de los datos, o sea, el porcentaje de datos que se mantenía igual a los originales.

Se describirá a continuación una de las pruebas realizadas con una generación del 10% de valores ausentes y luego se comentarán los resultados generales obtenidos.

La generación del 10% de valores ausentes se hizo sobre la base de datos original. Se generaron dos números aleatorios, el primero entre uno y la cantidad de filas y el otro entre dos y la cantidad de campos. Estos números constituían la fila y la columna respectivamente donde se insertaba un valor ausente.

Después de seleccionar la base de datos con la herramienta, se realizó el conteo de ausentes de cada uno de los campos y se obtuvo la información que se muestra en la tabla 3.2.

Tabla 3.2 Información sobre los valores ausentes

Atributo	Cantidad de ausentes	Cantidad de cadenas vacías	Cantidad de ceros	% de Valores Ausentes
age	429	0	0	9.0%
fnlwgt	444	0	0	10.0%
workclass	493	0	0	11.0%
education	437	0	0	10.0%
education-num	464	0	0	10.0%
marital-status	487	0	0	11.0%
ocupation	436	0	0	10.0%
race	444	0	0	10.0%
relationships	472	0	0	10.0%
sex	488	0	0	11.0%
hours-per-week	495	0	0	11.0%
Id	0	0	0	0.0%
native-country	493	0	0	11.0%

Es necesario observar que la aleatoriedad hizo que el porcentaje de ausentes en cada uno de los atributos fuera también alrededor del 10%.

En la herramienta se muestra la cantidad de cadenas vacías y cantidad de ceros en los atributos, pues como se ha dicho, estos pueden constituir también valores ausentes a considerar; como en este caso no se reportan, la herramienta solo tendrá en cuenta los valores ausentes propiamente dichos.

En este mismo paso se obtuvieron los patrones de valores ausentes, en total resultaron 173 patrones. Algunos de los patrones que más se repiten se muestran en la tabla 3.3.

Tabla 3.3 Patrones más frecuentes de valores ausentes

Patrón	Cantidad	Porcentaje de ausentes
1101111111111	171	0.08%
1111101111111	168	0.08%
1111111110111	152	0.08%
1111111101111	150	0.08%
1111111111011	148	0.08%
1111111111110	143	0.08%
1111110111111	142	0.08%
0111111111111	142	0.08%
1110111111111	141	0.08%
1111111011111	139	0.08%
1011111111111	139	0.08%
1111011111111	134	0.08%

Observar que son patrones con un solo valor ausente en cada campo.

Los patrones con mayor cantidad de ausentes se muestran en la tabla 3.4.

Tabla 3.4 Patrones con mayor cantidad de valores ausentes

Patrón	Cantidad	Porcentaje de ausentes
0010001110011	1	0.54%
1100110101010	1	0.46%
0110010011110	1	0.46%
0111000110110	1	0.46%
1000110101110	1	0.46%
1100000011111	1	0.46%
0010110100111	1	0.46%

Los patrones anteriores contienen seis o siete valores faltantes, pero a cada patrón solo pertenece un solo registro para un total de siete registros.

El siguiente paso de utilizar casos completos, no se tuvo en cuenta pues se eliminaría un gran número de registros y esto siempre lleva a perder datos válidos en el resultado.

En la ejecución del paso tres se tomó el umbral de usabilidad UU=0.4% de valores ausentes y con ello se eliminaron los 7 registros, correspondientes a los patrones mostrados en la tabla 3.4. Este es un número pequeño si se tiene en cuenta el total de registros.

El siguiente paso del proceso consiste en buscar algún campo que pueda ser reemplazado por el valor de otro campo de la misma tabla, por una expresión, o por algún valor de alguna otra tabla. En este caso se tuvieron en cuenta los atributos education y education-num, los cuales pueden ser obtenidos uno del otro, como se muestra en la tabla 3.5.

Tabla 3.5 Correspondencia entre education y education-num

education	education-num
Bachelors	13
Doctorate	16
Masters	14
Some-college	10
…	…

Esto garantiza que si existe un dato, el otro se puede imputar con exactitud, solo quedarían pendientes aquellos casos en que los dos estuvieran ausentes, y según los patrones de ausentes, esto ocurre solamente en 41 registros, como se muestra en la tabla 3.6.

Tabla 3.6 Patrones con valores ausentes en los atributos education y education-num

Patrón	Cantidad	Porcentaje de ausentes
0010001110011	1	0.54%
0100011111111	1	0.31%
0110010011110	1	0.46%
0110010111011	1	0.38%
0110010111111	1	0.31%
0110011111011	1	0.31%
0110011111111	3	0.23%
1010001111110	1	0.38%
1010010110111	1	0.38%
1010011111111	2	0.23%
1100000011111	1	0.46%
1100011011111	1	0.31%
1100011111110	1	0.31%
1110011011110	1	0.31%
1110011100111	1	0.31%
1110011101011	1	0.31%
1110011101111	1	0.23%
1110011110111	1	0.23%
1110011111011	1	0.23%
1110011111111	19	0.15%
Total	41	

A continuación se realizó la imputación a partir de los estadígrafos fundamentales. En el caso de los atributos numéricos, se usaron indistintamente la media, mediana, moda y desviación estándar. En el caso de los atributos discretos, se utilizaron la moda y el método de mantener las frecuencias de aparición propuesto en el trabajo.

En uno de los casos, en lugar de usar los estadígrafos, se intentó realizar el siguiente paso de aplicar la imputación por regresión lineal, pero no se encontraron dependencias lineales entre los atributos y se utilizaron los estadígrafos de tendencia central y de dispersión.

Con este procesamiento se imputaron todos los valores ausentes de la base de datos y se obtuvieron en las tres pruebas la cantidad de valores coincidentes con los datos originales para cada atributo que se muestra en la tabla 3.7.

Tabla 3.7 Valores correctos después de la imputación en cada prueba

Atributo	Prueba 1 Exactos	Prueba 2 Exactos	Prueba 3 Exactos
age	4161	4245	4178
fnlwgt	4132	4167	4145
workclass	4357	4386	4258
education	4560	4539	4558
education-num	4537	4543	4539
marital-status	4261	4212	4323
ocupation	4188	4239	4165
race	4455	4378	4269
relationships	4227	4265	4283
sex	4359	4380	4361
hours-per-week	4083	4123	4117
native-country	4484	4401	4452
TOTAL	51804	51878	51648
Exactitud	94,40	94,53	94,11

Los porcentajes de exactitud obtenidos, cuando se han imputado el 10% de los valores ausentes, están alrededor del 94% por lo que las imputaciones realizadas no degradaron significativamente los datos originales.

Al realizar las pruebas con el 5%, 10%, 15% y 20% se obtuvieron los resultados que se muestran en la tabla 3.8.

Tabla 3.8 Exactitud obtenida en las pruebas

% valores ausentes	Prueba	Exactitud
5%	1	97,6%
	2	97,3%
	3	97,4%
10%	1	94,4%
	2	94,5%
	3	94,1%
15%	1	88,2%
	2	89,1%
	3	88,7%
20%	1	83,2%
	2	82,9%
	3	83,1%

En la tabla 3.8 la exactitud es menor en la medida que la cantidad de valores ausentes aumenta, esto es de esperar pues hay un valor mayor de imputaciones a realizar.

Aunque la herramienta, en su versión actual, no contempla todos los métodos descritos en el procedimiento, brinda un método de trabajo que de forma interactiva permite a un usuario-analista de datos hacer las sustituciones de los valores ausentes. No siempre hay valores ausentes en todos los campos (como en los casos que se ha probado), lo que hace más fácil el uso del procedimiento descrito.

3.2. Conclusiones parciales

Se cuenta con cuatro herramientas para la realización de las tareas de análisis, reemplazo de valores ausentes, estandarización de cadenas de textos y segmentación de direcciones postales, que contribuyen al proceso de Limpieza de datos y evitan tener que adquirir otras, que pudieran no adaptarse completamente a nuestras condiciones o requerir de grandes inversiones.

A partir de las pruebas y estudios realizados con las herramientas creadas, en las bases de datos de diferentes empresas e instituciones de nuestro territorio, se puede constatar que las mismas ayudan a garantizar la calidad de los datos que almacenan, ya que su aplicación arrojó resultados satisfactorios.

Conclusiones

Una vez culminada la investigación, se pueden plantear las siguientes conclusiones:

- Se obtuvo una taxonomía de errores que corrobora para nuestro entorno lo que se reporta en la literatura y sirvió de base para decidir los problemas más importantes a tratar

- Se definieron las distancias EDUK y d_{TOK}, que en el caso de las cadenas con errores tipográficos resultan buenas para formar grupos de cadenas semejantes que facilitan la estandarización de las mismas. Estas distancias muestran un comportamiento similar a otras distancias en la estandarización de cadenas con errores tipográficos

- Se diseñó un marco de trabajo basado en técnicas de agrupamiento para la estandarización de cadenas de caracteres que fue el fundamento para la construcción de una herramienta cuya aplicación resultó satisfactoria.

- Se diseñó e implementó un procedimiento general para la sustitución de valores ausentes obteniéndose resultados satisfactorios en su aplicación.

- Se definió un algoritmo para la imputación de valores ausentes de dominios discretos, que pudiera ser usado cuando otros estadígrafos no sean aplicables y que permitió lograr la sustitución de los valores ausentes, manteniendo en el conjunto de datos la frecuencia de los valores válidos.

- Las herramientas obtenidas brindan una solución metodológica y práctica al problema de la limpieza de datos y su utilización permitió validar su efectividad para el entorno cubano.

Recomendaciones

- Realizar un estudio sobre el valor máximo a tomar de la distancia entre cadenas, que se utiliza en la búsqueda en los diccionarios, para considerar dos cadenas semejantes, en el marco de la estandarización de las direcciones postales
- Estudiar, con vistas a su incorporación al Asistente para el reemplazo de valores ausentes, las reglas de asociación pues a partir de las mismas se pueden descubrir relaciones entre atributos que pudieran usarse para determinar los valores de los ausentes en los campos de los registros.
- Incorporar al Asistente para el reemplazo de los valores ausentes los métodos que se describen en el paso ocho del procedimiento y que actualmente no se han implementado.
- Integrar las herramientas en una suite para realizar la Limpieza de datos.

Referencias bibliográficas

1. Agrawal, R. and R. Srikant. *Fast Algorithms for Mining Association Rules in large Database* in *20th International Conference on Very Large Data Bases*. 1994. San Francisco: Morgan Kaufmann Publishers Inc.

2. Amon, I. and C. Jimenez, *Hacia una metodología para la selección de técnicas de depuración de datos.* Revista Avances en Sistemas e Informática. 6(1): p. 185-190, 2009.

3. Amy, F. and C. Zhengxin. *Duplicate detection using k-way sorting method* in *Proceedings of the 2000 ACM symposium on Applied computing - Volume 1*. 2000. Como, Italy: ACM.

4. Analystics Stones, *Second Moment. The news and business resource for applied analystics. Missing Data Method*. 2003, disponible en: http://www.secondmoment.org/etal-column/index.php, último acceso: 23/05/2009.

5. Ananthakrishna, R., S. Chaudhuri, and V. Ganti. *Eliminating Fuzzy Duplicates in Data Warehouses* in *Proceedings of the 28th VLDB Conference*. 2002. Hong Kong,China.

6. Anónimo, *Information Technology Services. Handling Missing or Incomplete Data*. 2004, disponible en: http://www.utexas.edu/its/rc/answers/general/gen25.html, último acceso: 08/07/2007.

7. Anónimo, *Data Cleansing and Warehouses with XML*. 2005, disponible en: http://www.infonyte.com/en/solutions.html, último acceso: 20/05/2006.

8. Arasu, A., S. Chaudhuri, and R. Kaushik. *Learning String Trasnformations From Examples* in *VLDB' 09*. 2009. Lyon, France.

9. Ballard, C., et al., *Dimensional Modeling: In a Business Intelligence Environment*, USA: International Business Machines Corporation, 2006.

10. Bernstein, P.A. and L.M. Hass, *A guide to the tools and core technologies for merging information from disparate sources.* Communications of the ACM. 51(8): p. 72-79, 2008.

11. Berti-Équille, L. and D. Tamraparni. *Data Quality Mining: New Research Directions* in *Proceedings of ICDM 2009*. 2009. Miami.

12. Bilenko, M. and R.J. Mooney, *On evaluation and training-set construction for duplicate detection.* Proc. of the KDD-2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation: p. 7-12, 2003.

13. Borkar, V., K. Deshmukh, and S. Sarawagi, *Automatic segmentation of text into structured records.* SIGMOD‹ 01, 2001.

14. Brachman, R.J. and A. Tej, *The process of knowledge discovery in databases*, in *Advances in knowledge discovery and data mining*, American Association for Artificial Intelligence. p. 37-57. 1996.

15. Cadot, M. and J. di-Martino, *A data cleaning solution by Perl scripts for the KDD Cup 2003 task 2.* SIGKDD Explorations. 5(2): p. 158 - 159, 2003.

16. Campan, A., G. Serban, and A. Marcus, *Relational Association rules and error detection.* Informatica. LI(1), 2006.

17. Cañizares, M., I. Barroso, and K. Alfonso, *Datos incompletos: una mirada crítica para su manejo.* Gaceta Sanitaria. 18(1), 2004.

18. Cohen, W.W., P. Ravikumar, and S.E. Fienberg. *A Comparison of String Distance Metrics for Name-Matching Tasks* in *Proceedings of II Web*. 2003.

19. Cohen, W.W. and J. Richman, *Learning to match and cluster Entity names.*

20. Cong, G., et al. *Improving Data Quality: Consistency and Accuracy* in *VLDB'07*. 2007. Vienna, Austria.

21. Coppola, L., et al., *Bayesian networks for imputation in official statistics: a case study.* DataClean Conference: p. 30-31, 2000.

22. Chan, S.K., B. He, and I. Ounis. *An in-depth study of the automatic detection and correction of spelling mistakes* in *5th Dutch-Belgian Information Retrival Workshop*. 2005.

23. Chapman, A.D., *Principles and Methods of Data Cleaning. Primary species and species ocurrence data. Version 1.0.* Report for the Global Biodiversity Information Facility. Copenhagen, 2005.

24. Chaudhuri, S., et al., *Data Cleaning in Microsoft SQL Server 2005.* SIGMOD, 2005.

25. Chaudhuri, S., et al., *Robust and efficient fuzzy match for online data cleaning.* Proceedings of ACM SIGMOD International Conference on Management of Data: p. 313-324, 2003.

26. Chaudhuri, S., V. Ganti, and D. Xin, *Exploiting web search to generate synonyms for entities*. Proceedings of the 18th international conference on World wide web, Madrid, Spain: ACM. 151-160, 2009.

27. Chaudhuri, S.R., V.B. Ganti, and R. Ananthakrishna, Detecting duplicate records in databases, Número de patente: US 7,685,090 B2, USA. 2010

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

29. Christen, P. *Febrl - An Open Source Data Cleaning, Deduplication and Record Linkage System with a Graphical User Interface* in *KDD'08*. 2008. Las Vegas, Nevada, USA.

30. Christen, P., *Development and User Experiences of an Open Source Data Cleaning, Deduplication and Record Linkage System.* SIGKDD Explorations. 11(1): p. 39-48, 2009.

31. Damerau, F.J., *A technique for computer detection and correction of spelling errors*. Communication of ACM. 7(3), 1964.

32. Dasu, T., G.T. Vesonder, and J.R. Wright. *Data Quality through knowledge engineering* in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2003. Washington, D.C.

33. Dermeit, M.M., R. Funk, and M. Dennis, *Data cleaning and replacements of missing values*. 1999, disponible en: <http://www.chestnut.org/li/downloads/training_memos/>, último acceso: 22/10/2010.

34. Deza, M.M. and E. Deza, *Dictionary of distances*, Elsevier. 2006.

35. Elmagarmid, A. and P.G. Verykios, *Duplicate Record Detection: A Survey*. Knowledge and Data Engineering IEEE Transactions on Publication. 19: p. 1-16, 2007.

36. Farhangfar, A., L. Kurgan, and W. Pedrycz, *Experimental analysis of methods for handling missing values in databases.* Intelligent Computing: Theory and Applications II, 2004.

37. Foley, M.J., *Preemptive Data Cleaning: Techniques.* SUGI 26 Proceedings Tutorials: p. 15-26, 2006.

38. Fox, C., A. Levitin, and T. Redman, *The Notion of Data and Its Quality Dimensions.* Information Processing and Management. 30(1): p. 9-19, 1994.

39. Gadd, T.N., *PHONIX: The algorithm.* Program: Automated Library and Information Science. 24(4): p. 363 - 366, 1988.

40. Galhardas, H., *Data Cleaning and Integration. Data Cleaning Commercial Tools*. 2001, disponible en: http://web.tagus.ist.utl.pt/~helena.galhardas/cleaning.html, último acceso: 24/07/2010.

41. Galhardas, H., et al. *AJAX: an extensible Data Cleaning Tool* in *Proc. of ACM SIGMOD Conf. on Management of Data.* 2000.

42. Galhardas, H., et al. *Declarative Data Cleaning: Language, Model and Algorithms* in *VLDB - International Conference on Very Large Databases*. 2001: Morgan Kaufmann.

43. Gálvez, C., *Aplicación de transductores de estado-finito a los procesos de unificación de términos.* Ci. Inf. 35(3): p. 67-74, 2006.

44. Gálvez, C., *Identificación de nombres personales por medios de sistemas de codificación fonética.* Encontros Bibli. Segundo Semestre(022): p. 105-116, 2006.

45. Gálvez, C. and F. Moya-Anegón, *Approximate Personal Name-Matching Through Finite-State Graphs*. 2007, disponible en: www.interscience.wiley.com, último acceso: 09/09/2011.

46. Garson, G.D., *Data Imputation for Missing Value*. 2006, disponible en: http://www2.chass.ncsu.edu/garson/pa765/missing.htm, último acceso: 20/09/2007.

47. Gartner, *Dirty data is a Business Problem, Not an IT Problem*. 2007, disponible en: http://www.gartner.com/it/page.jsp?id=501733, último acceso: 03/09/2011.

48. Gelbukh, A., et al. *Generalized Monge-Elkan method for approximate text string comparision* in *10th International Conference on Computational Linguisticsand Intelligent Text Processing*. 2009.

49. Graham, J.W., *Missing Data Analysis: Making It Work in the Real World.* Annu. Rev. Psychol. 60: p. 549-576, 2009.

50. Greenfield, L., *What data errors you may find when building a data warehouse*. 2007, disponible en: www.dwinfocenter.org., último acceso: 05/09/2011.

51. Guyon, I., N. Matic, and V. Vapnik, *Discovering informative patterns and data cleaning*, in *Advances in knowledge discovery and data mining*, American Association for Artificial Intelligence. p. 181-203. 1996.

52. Hall, P. and G. Dowling, *Approximate string matching.* ACM Computing Surveys. 12(4): p. 381-402, 1980.

53. Hall, R., C. Sutton, and A. McCallum. *Unsupervised Deduplication using Cross Field Dependences* in *KDD'08*. 2008. Las Vegas, Nevada, USA: ACM.

54. Hawkins, S., et al., *Outlier Detection Using Replicator Neural Networks*. Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery: Springer-Verlag. 170-180, 2002.

55. Hernández, M.A. and S.J. Stolfo, *Real world Data is Dirty: Data Cleansing and The Merge-Purge Problem.* Journal of data mining and Knowlwgde Discovery. 2(1), 1998.

56. Hobbs, L., et al., *Oracle Database 10g Data Warehousing*, USA: Elsevier Digital Press, 2005.

57. Hu, M., S.M. Salvucci, and M.P. Cohen, *Evaluation of some popular imputation algorithms.* The Survey Research Methods Section of the ASA: p. 308-313, 1998.

58. Inmon, W.H., *Building the Data Warehouse.* Fourth Edition ed, Indianapolis, USA: Wiley Publisher, 2005.

59. Jebamalar-Tamilselvi, J. and V. Saravanan, *A Unified Framework and Sequential Data Cleaning Approach for a Data Warehouse.* International Journal of Computer Science and Network Security. 8(5), 2008.

60. Jin, L., C. Li, and S. Mehrotra, *Efficient Record Linkage in Large Data Sets.* DASFAA '03: Proceedings of the Eighth International Conference on Database Systems for Advanced Applications: p. 137, 2003.

61. Johnson, T., *Data Quality and Data Cleaning: An Overview.* Lecture notes for CS541, 2004.

62. Kaleen, A., et al. *Address Standardization using Supervised Machine Learning* in *International Conference on Computer Communication and Management*. 2011. Singapore.

63. Kaufman, L. and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.*, New York: John Wiley, 1990.

64. Kedad, Z. and E. Métais, *Ontology-Based Data Cleaning*. 2002, disponible en: wotan.liu.edu/docis/dbl/nldbnl/2002__137_ODC.htm, último acceso: 01/03/2004.

65. Kim, W., et al., *A Taxonomy of Dirty Data.* Data Min. Knowl. Discov. 7, 2003.

66. Kimball, R., *Dealing with Dirty Data*. 1996, disponible en: www.dbmsmag.com/9609d14.html, último acceso: 03/09/2008.

67. Kimball, R. and J. Caserta, *The Data Warehouse ETL Toolkit. Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*, Indianapolis, USA: Wiley Publishing, Inc., 2004.

68. Knorr, E.M. and R.T. Ng, *Algorithms for Mining Distance-Based Outliers in Large Datasets.* Proc. 24th Int. Conf. Very Large Data Bases, VLDB: p. 392--403, 1998.

69. Kriegel, H.-P., P. Kröger, and A. Zimek, *Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Transactions on Knowledge Discovery from Data (TKDD)*. 3(1), 2009.

70. Kriegel, H.-P., et al., *Metric Spaces in Data Mining: Applications to Clustering.* The SIGSPATIAL Special. 2(2): p. 36-38, 2010.

71. Kukich, K., *Techniques for Automatically Correcting Words in Text.* Computing Surveys. 24(4): p. 377-439, 1992.

72. Lachev, T., *Applied Microsoft Analysis Services 2005 and Microsoft Business Intelligence Platform,* USA: Prologika Press, 2006.

73. Lakshminarayan, K., S.A. Harp, and T. Samad, *Imputation of missing data in industrial databases.* Applied Intelligence. 11(3): p. 259-275, 1999.

74. Lee, M.L., T.W. Ling, and W.L. Low, *IntelliClean: a knowledge-based intelligent data cleaner.* Proceedings of Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: p. 290-294, 2000.

75. Lee, M.L., et al., *Cleansing data for mining and datawarehousing.* p. 10, 1999.

76. Lehti, P. and P. Fankhauser, *A Precise Blocking Method for Record Linkage.* Lecture Notes Computer Science. 3589: p. 210-220, 2005.

77. Levenshtein, V., *Binary Codes capable of correcting deletions, insertions and reversals.* Soviet Physics Doklady. 10(8): p. 707-710, 1966.

78. Levenshtein, V.I., *Binary Code capable of correcting deletions, insertions and reversal.* Soviet Physics Doklady. 163(4): p. 845-848, 1965.

79. Levitin, A. and T. Redman, *Model of the Data (Life) Cycles with Application to Quality.* Information and Software Technology. 35(4): p. 217-223, 1995.

80. Li, C., B. Wang, and X. Yang. *VGRAM: Improving Performance of Approximate Queries on String Collections Using Variable-Length Grams* in *VLDB'07*. 2007. Vienna, Austria.

81. Li, Z., et al., *A New Efficient Data Cleansing Method*. Proceedings of the 13th International Conference on Database and Expert Systems Applications: Springer-Verlag. 484-493, 2002.

82. Little, R.J. and D.B. Rubin, *Statistical Analysis with Missing Data*, New York: John Wiley and Sons, 1987.

83. López-Porrero, B. and R. Pérez-Vázquez, *El análisis de datos en la limpieza de datos*. Boletín de la Sociedad Cubana de Matématica y Computación. 3(1), 2005.

84. López-Porrero, B. and R. Pérez-Vázquez. *Estudio del comportamiento de valores ausentes en los datos de una empresa cubana* in *Evento de Informática de la V Conferencia Internacional de Ciencias Empresariales*. 2006. Santa Clara.

85. López-Porrero, B. and R. Pérez-Vázquez. *Un marco de trabajo para la estandarización de atributos tipo carácter* in *UCIENCIA 2007*. 2007.

86. López-Porrero, B. and R. Pérez-Vázquez, *Utilización de una extensión de la distancia de Levenshtein en la limpieza de datos*. Boletin de la Sociedad Cubana de Matemática y Computación. 5(Especial), 2007.

87. López-Porrero, B. and R. Pérez-Vázquez, *Monografía Limpieza de datos*, Samuel Feijó: Cuba. 2008.

88. López-Porrero, B. and R. Pérez-Vázquez, *Taxonomías de errores en las bases de datos cubanas.* Revista Cubana de Ciencias Informáticas. 2(1-2), 2008.

89. López-Porrero, B. and R. Pérez-Vázquez, *Estandarización de cadenas de texto en aplicaciones médicas.* Revista Cubana de Informática Médica. 9(1), 2009.

90. López-Porrero, B. and R. Pérez-Vázquez, *Reglas de asociación ordinales en la detección de errores en los datos.* Revista Cubana de Ciencias Informáticas. 4(1,2), 2010.

91. López-Porrero, B., et al. *La detección de errores en los datos, base de la obtención de un proceso con calidad en la toma de decisiones empresariales.* in *VI Conferencia internacional de Ciencias Empresariales*. 2007. Villa Clara, Cuba.

92. López-Porrero, B., R. Pérez-Vázquez, and B. Pimentel-Martell, *Validación de la herramienta de análisis de datos DbAnalyzer*, in *Memorias del evento Informatica Empresarial de la IV Semana tecnológica*: Ciudad de la Habana. p. 15. 2006.

93. Loshin, D., *The Practitioner's Guide to Data Quality Improvement.* Editorial Elsvier, 2011.

94. Maletic, J.I. and A. Marcus, *Progress Report on Automated Data Cleansing.* Technical Report CS-99-02 Kent State University, 1999.

95. Maletic, J.I. and A. Marcus, *Automated Identification of Errors in Data Sets.* Technical Report CS-00-02. Kent State University, 2000.

96. Maletic, J.I. and A. Marcus. *Data Cleansing: Beyond Integrity Analysis* in *Conference on Information Quality (IQ2000)*. 2000. Massachusetts Institute of Technology.

97. Maletic, J.I. and A. Marcus, *Utilizing Association Rules for Identification of Possible Errors in Data Sets.* Technical Report CS-00-03. Kent State University., 2000.

98. Marcus, A. and J.I. Maletic, *Cap. 2 A Prelude to Knowledge Discovery. The Data Mining and Knowledge Discovery Handbook*, ed. O.M.a.L. Rokach: Springer, 2005.

99. Marmol-Lacal, D. and B. López-Porrero, *Determinación de una taxonomía de errores en los sistemas operacionales de nuestro entorno*, in *Departamento Ciencia de la Computación*, Universidad Central de las Villas: Santa Clara. 2005.

100. Martín, Q. and M.T. Cabero-Moran, *Tratamiento estadístico de datos con SPSS*, España: Paraninfo, 2008.

101. Monge, A.E. and C.P. Elkan, *The field matching problem: Algorithms and applications.* In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining: p. 267--270, 1996.

102. Monge, A.E. and C.P. Elkan, *An efficient domain-independent algorithm for detecting approximately duplicate database tuples.* Proceedings of the SIGMOD 1997 workshop on data mining and knowledge discovery, 1997.

103. Moreau, E., F. Yvon, and O. Cappé. *Robust similarity measures for named entities matching.* in *22nd International Conference on Computational Linguistics*. 2008.

104. Müller, H. and J.C. Freytag, *Problems, Methods, and Challenges in Comprehensive Data Cleansing.* Technical Report HUB-IB-164, Humboldt University Berlin, 2003.

105. Mundy, J., W. Thornthwaite, and R. Kimball, *The Microsoft Data Warehouse Toolkit: With SQL Server 2005 and the Microsoft Business Intelligence Toolset*, Indianapolis, USA: Wiley Publishing, Inc., 2006.

106. Navarro, G., *A guided tour to approximate string matching.* ACM Comput. Survey. 33: p. 31-88, 2001.

107. Oliveira, P., et al. *A taxonomy of Data quality Problems* in *DIQ' 05- 2nd International worshop on data and information quality*. 2005. Porto Portugal.

108. Olson, J.E., *Data Quality the accuracy dimension*: Morgan Kaufmann. 293, 2003.

109. Orr, K., *Data Quality and Systems Theory.* Communications of the ACM. 41(2): p. 66-71, 1998.

110. Philips, L., *The double metaphone search algorithm.* C/C++ User´s Journal. 18(6), 2000.

111. Piskorki, J. and M. Sydow, *Usability of string distance metrics for names matching task in Polish.* Proc. of LTC'07: p. 403-407, 2007.

112. Pyle, D., *Data Preparation for Data Mining*, San Francisco, California: Morgan Kaufmann Publishers, 1999.

113. Quinlan, J.R., *Induction of decision trees.* Maching learning. 1(1): p. 81-106, 1986.

114. Quinlan, J.R., *C4.5 Programs for maching learninng*, San Francisco CA USA: Morgan Kaufmann Publishers Inc., 1993.

115. Rahm, E. and H.H. Do, *Data Cleaning: Problems and Current Approaches.* IEEE. Bulletin of the Technical Committee on Data Engineering. 23(4): p. 3-13, 2000.

116. Raman, V. and J.M. Hellerstein, *Potter's Wheel: An Interactive Data Cleaning System.* The VLDB Journal: p. 381-390, 2001.

117. Ramirez, F. and E. López. *Spelling error patterns in spanish for word processing applications* in *Fifth internacional conference on Language resources and evaluation, LREC*. 2006.

118. Redman, T., *The Impact of Poor Data Quality on the Typical Enterprise.* Commun. ACM. 41: p. 79-82, 1998.

119. Rema, A., et al., *Rule based synonyms for entity extraction from noisy text*. Proceedings of the second workshop on Analytics for noisy unstructured text data, Singapore: ACM. 31-38, 2008.

120. Reuther, P., *Personal Name Matching: New Test Collections and a Social Network based Approach.* Trierer Forschungsberichte Mathematik / Informatik. 1, 2006.

121. Reuther, P. and B. Walter, *Survey on test collections and techniques for personal name matching.* International Journal of Metadata, Semantics and Ontologies. 1(2): p. 89-99, 2006.

122. Rubin, D.B., *Multiple imputation for nonresponse in surveys*, New York: John Wiley & Sons, 1987.

123. Rubin, D.B., *Multiple Imputation after 18+year.* Journal of the American Statistical Association. 91, 1996.

124. Sarmad, H., G. Sana, and W. Afifah, *Developing lexicographic sorting: An example for Urdu.* ACM Transactions on Asian Language Information Processing (TALIP). 6(3): p. 10, 2007.

125. Sattler, K.U. and E. Schallehn, *A Data Preparation Framework based on a Multidatabase Language.* International Database Engineering Applications Symposium (IDEAS), 2001.

126. Sauleau, E.A., J.P. Paumier, and A. Buemi, *Medical record linkage in health information systems by approximate string matching and clustering.* BMC Medical Informatics and Decision Making. 5(32), 2005.

127. Schafer, J.L., *Analysis of Incomplete Multivariate Data*, London: Chapman and Hall. 430 pp., 1997.

128. Scherbina, A., *Clustering of Web Access Sessions.* Lecture Notes in Computer Science, 2005.

129. Sleit, A., et al., *Applying Ordinal Association Rules for Cleansing Data With Missing Values.* Journal of American Science. 5(3): p. 52-62, 2009.

130. Smith, T.F. and M.S. Waterman, *Identification of common molecular subsequences.* Journal Molecular Biology, (147): p. 195-197, 1981.

131. Sung, S.Y., Z. Li, and P. Sun, *A fast filtering scheme for large database cleansing.* CIKM '02: Proceedings of Eleventh ACM International Conference on Information and Knowledge Management, 2002.

132. Suzanne, M.E., et al., *Adapting integrity enforcement techniques for data reconciliation.* Inf. Syst. 26(8): p. 657-689, 2001.

133. Ukkonen, *Approximate string matching with q-grams and maximal matches.* Theoretical Computer Science, (1): p. 191-211, 1992.

134. Usama, M.F., *Mining Database: Towards Algorithms for Knowledge Discovery.* IEEE Techn. Bulletin Data Engineering. 21(1), 1998.

135. Usama, M.F., P.-S. Gregory, and U. Ramasamy, *Summary from the KDD-03 panel: data mining: the next 10 years.* SIGKDD Explor. Newsl. 5(2): p. 191-196, 2003.

136. Usama, M.F., G. Piatetsky-Shapiro, and P. Smyth, *From data mining to knowledge discovery: an overview*, in *Advances in knowledge discovery and data mining*, American Association for Artificial Intelligence. p. 1-34. 1996.

137. Vassiliadis, P., A. Simitsis, and E. B. *A taxonomy of ETL activities* in *DOLAP' 09 Proceeding of the ACM twelfth international workshop on Data warehousing and OLAP*. 2009. NY, USA.

138. Wang, R., M. Ziad, and Y.W. Lee, *Data Quality*: Kluwer Academic Publishers, 2001.

139. Winkler, W.E., *Overview of record linkage and current research directions.* Technical Report RR2006/02, 2006.

140. Wong, W., W. Liu, and M. Bennamoun, *Enhanced Integrated Scoring for Cleaning Dirty Texts.* 2008.

141. Wright, R.N. and G. Jagannathan, *Privacy-preserving imputation of missing data.* Data & Knowledge Engineering. 65: p. 40-56, 2008.

142. Yu, D., S. Sheikholeslami, and A. Zhang, *FindOut: Finding Outliers in Very Large Datasets.* Knowledge and Information Systems. 4(4): p. 387-412, 2002.

143. Zobel, J. and P. Dart. *Phonetic String Matching: Lessons from Information Retrieval* in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. 1996.

ANEXO A

Resumen de las herramientas de Limpieza de datos

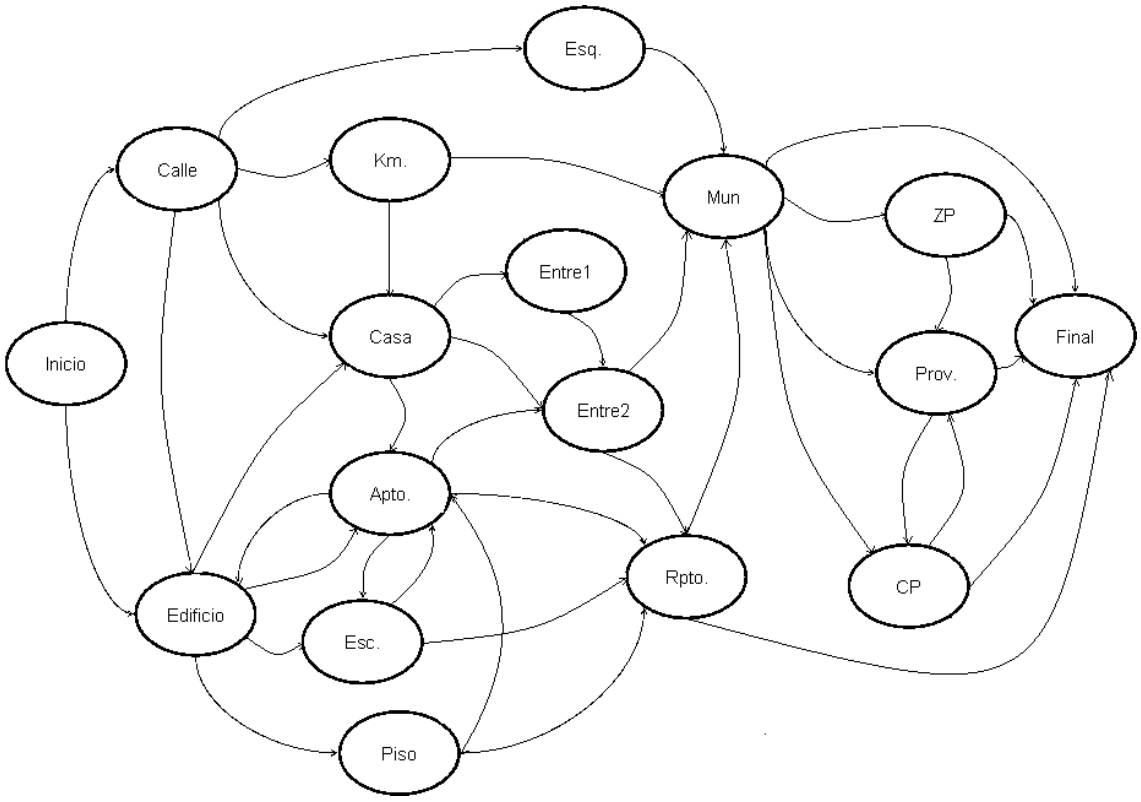
Herramienta	Compañía que la produce	Objetivo	Funcionalidades
Trillium Software System	Trillium Software	Corrige y estandariza lista de nombres y direcciones.	Análisis, estandarización, Eliminación de duplicados
MatchMarker	Info Tech ltd	Limpia y estandariza direcciones	Correcciones ortográficas, eliminación de duplicados
QuickAdress Batch	QAS Systems	Limpia y estandariza direcciones	Correcciones ortográficas, sustituye valores ausentes, elimina duplicados
PureName PureAddress	Carleton	Limpia nombres y direcciones	Análisis, estandarización
SQL Server	Microsoft	Proceso de ETL	Eliminación de duplicados, transformaciones en general
Integrity	Validity	Eliminación de duplicados	Análisis e identificación de entidades.
Oracle 10 g	Oracle	Proceso de ETL	Eliminación de duplicados, transformaciones en general
ETI*Data Cleanse	Evolutionary Technologies International	Eliminación de duplicados	Estandariza, comparación de datos, encuentra duplicados
Centrus Merge/Purge	Quality marketing software	Detección de registros duplicados	Elimina registros duplicados en línea, integra información de clientes desde varias fuentes.
SSA-Name/Data Clustering Engine	Search Software America	Limpia nombres y direcciones	Resuelve errores ortográficos, y otros errores en nombres

			de personas y compañías mezcladas
DfPower	Dataflux Corporation	Detecta duplicados	Análisis, estandariza, compara registros y agrupa registros similares
MatchIT	helpIT Systems limited	Detecta duplicados	Corrección de direcciones, y detección de duplicados
d. Centric	firsLogic	Elimina registros duplicados	Parsing, corrección, estandarización, comparación de registros
reUnion and MasterMerge	PitneyBowes	Limpia datos y elimina registros duplicados	verifica, corrige y estandariza, identifica los registros duplicados
PureIntegrate	Carleton	Limpia datos y elimina registros duplicados	Resuelve errores en datos, inconsistencias, valores ausentes
TwinFinder	Omikron	Limpia datos y elimina registros duplicados	errores ortográficos, trasposiciones y abreviaturas
DoubleTake StyleList, Personator	Peoplesmith	Limpia datos y elimina registros duplicados	Divide nombres y direcciones, estandariza,
Data Tools Twins	Data Tools	Limpia y elimina duplicados	Analiza direcciones, nombres de estado y personas, estandariza direcciones.
NoDupes	Quess, Inc	Elimina duplicados	Limpia datos de dominio específico
DeDuce	The Computing Group	Elimina duplicados	Detecta duplicados manual y automáticamente , mezcla
DeDupe	International Software Publishing	Elimina duplicados	Usa operadores de correspondencia fonéticos, de palabras, etc.
Merge/Purge Plus	Group1 Software	Elimina duplicados	Limpia nombres y direcciones

Migration Architect	Evoke Software	Análisis de datos	Obtiene el perfil de los datos, desde los datos físicos.
WizWhy & WizRule	WizSoft Inc	Análisis de datos	Determina relaciones entre los datos, usando reglas de asociación.

ANEXO B

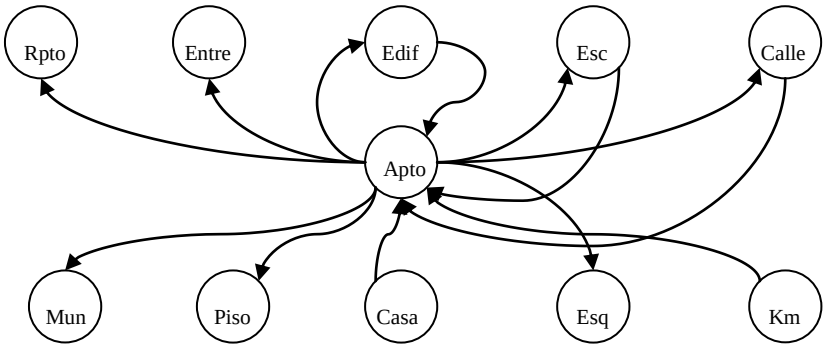
Grafo simplificado del HDD Externo



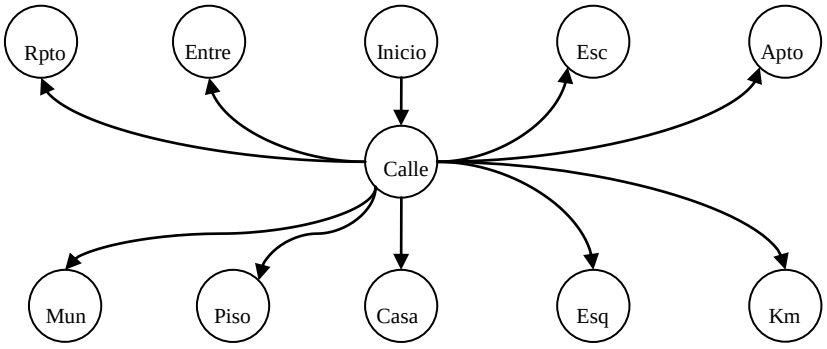
ANEXO C

Relaciones entre nodos del HDD externo

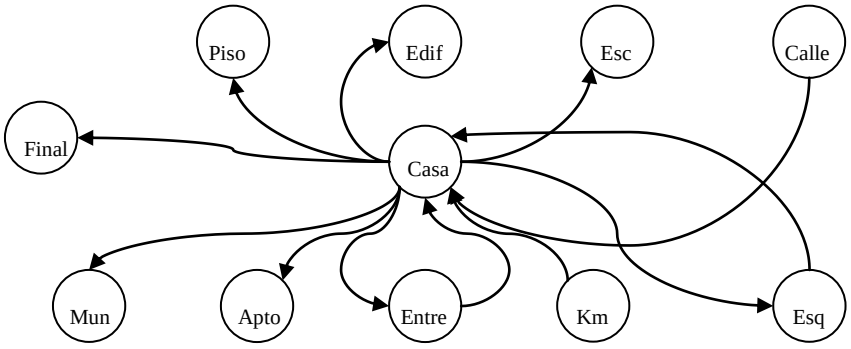
Relaciones de Apartamento



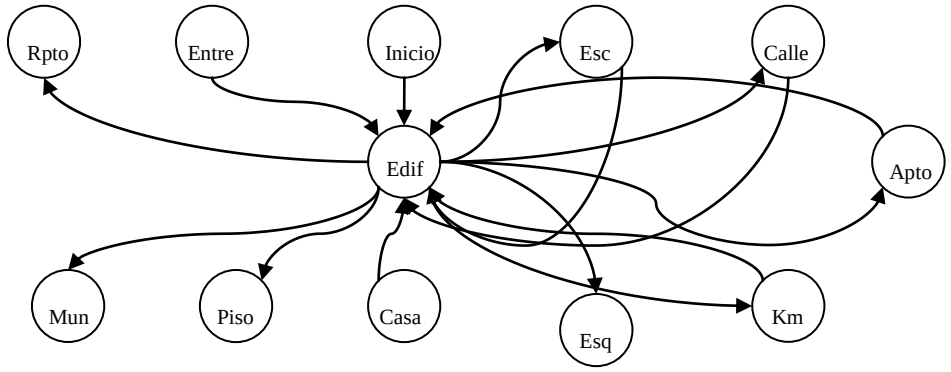
Relaciones de Calle



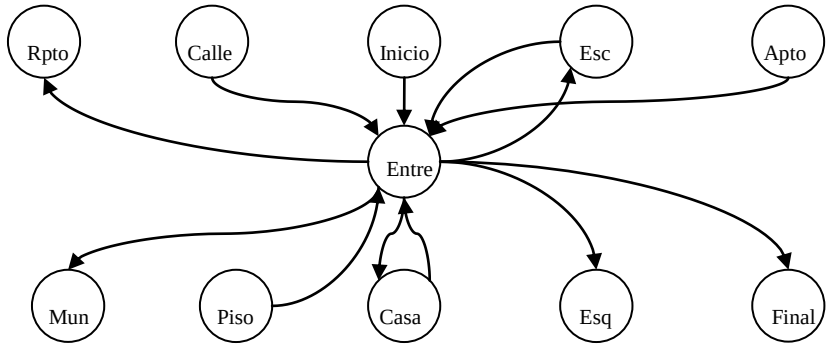
Relaciones de Casa



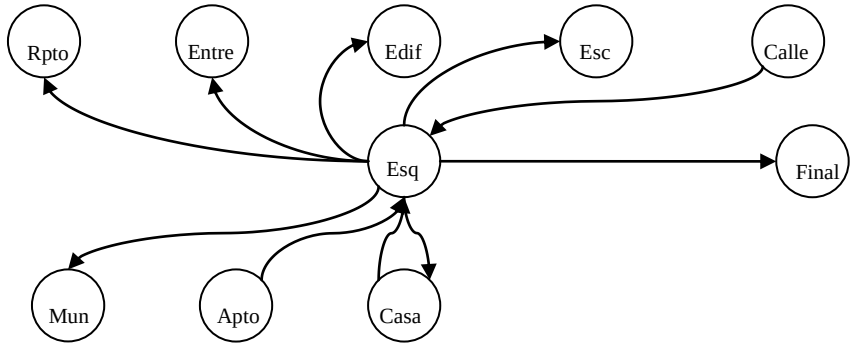
Relaciones de Edificio



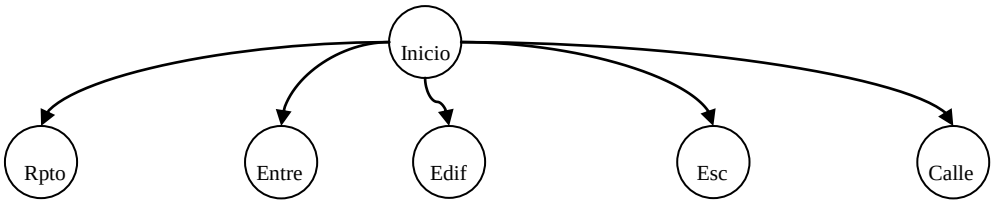
Relaciones de Entre calles



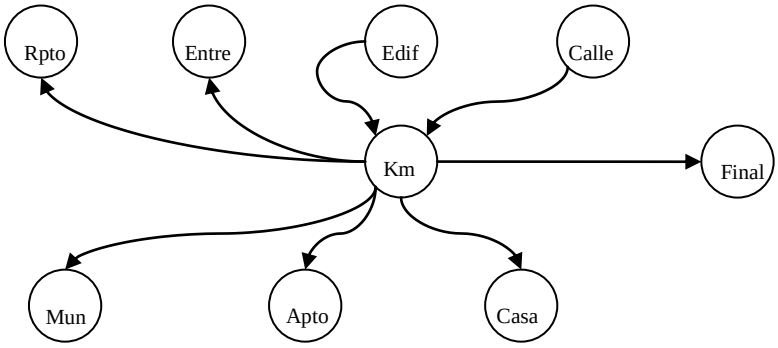
Relaciones de Esquina



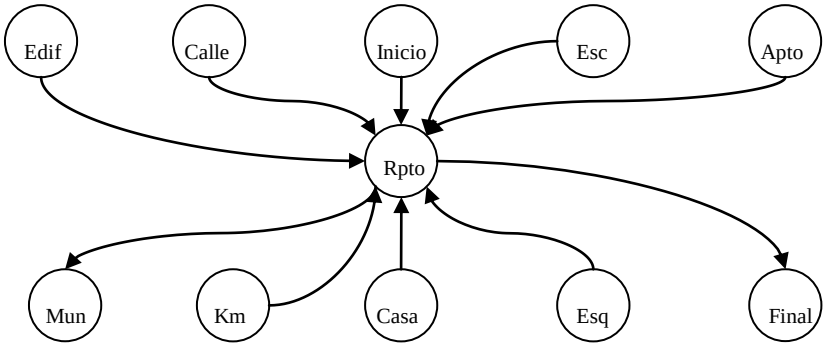
Relaciones de Inicio



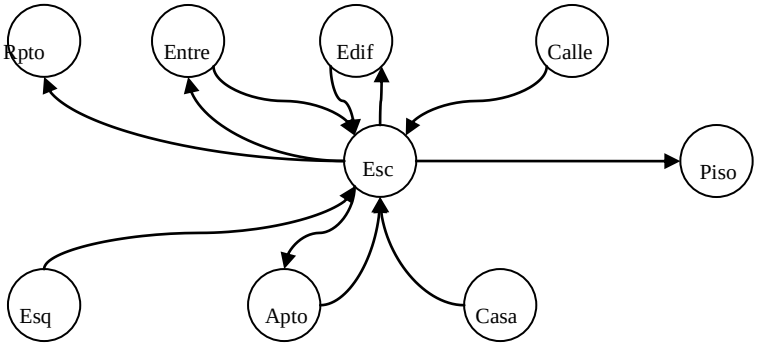
Relaciones de Kilómetro



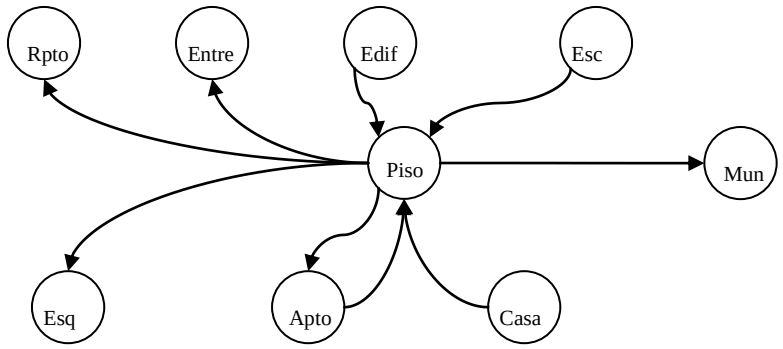
Relaciones de Reparto



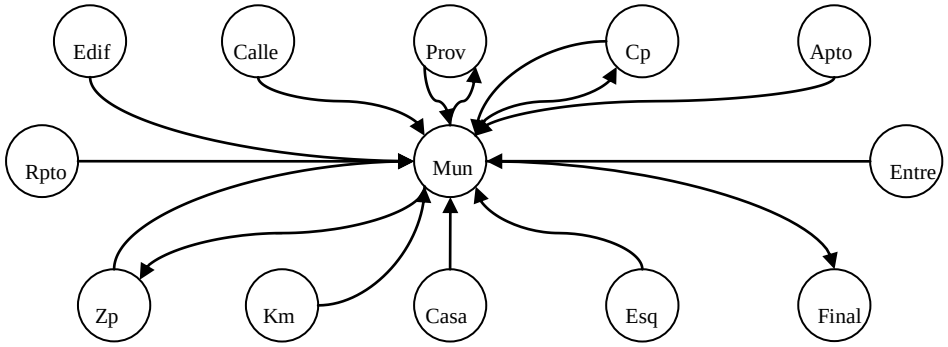
Relaciones de
Escalera



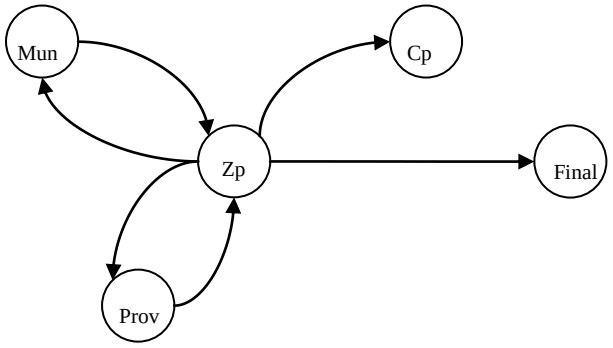
Relaciones de Piso



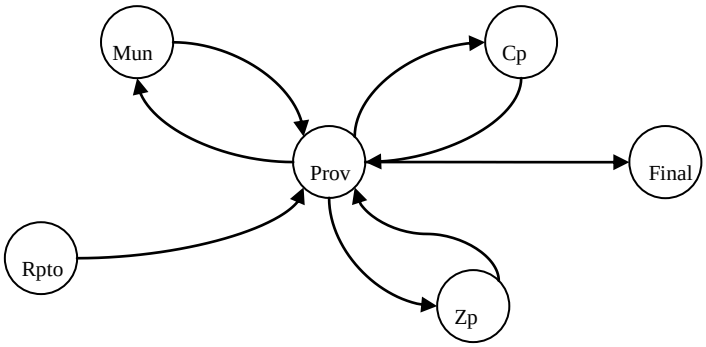
Relaciones de Municipio



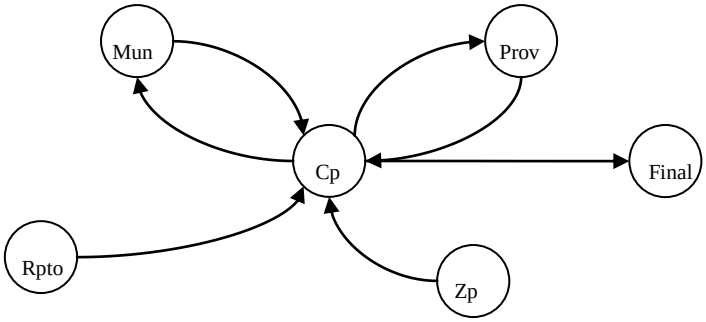
Relaciones de Zona postal



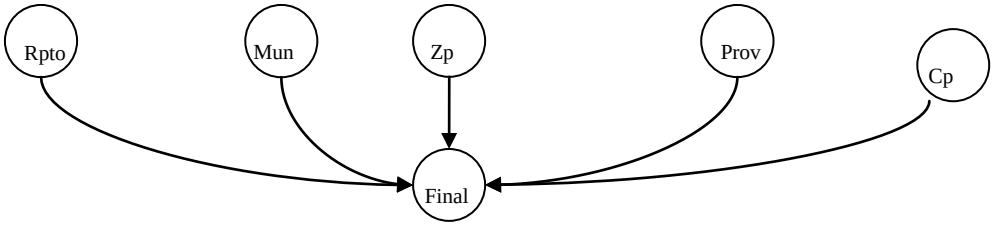
Relaciones de Provincia



Relaciones de Código postal



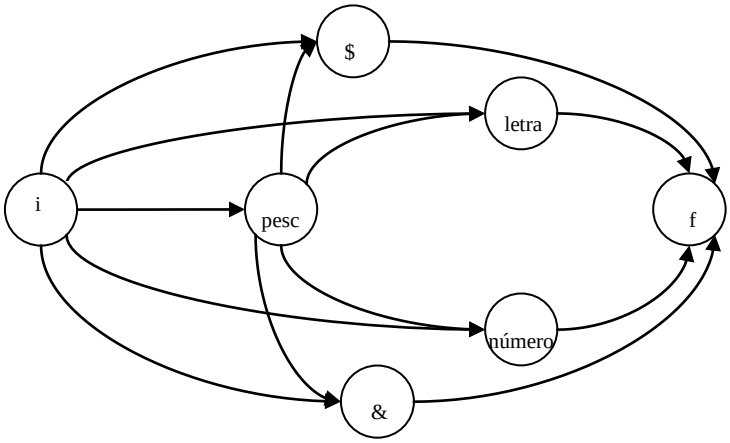
Relaciones de Final



ANEXO D

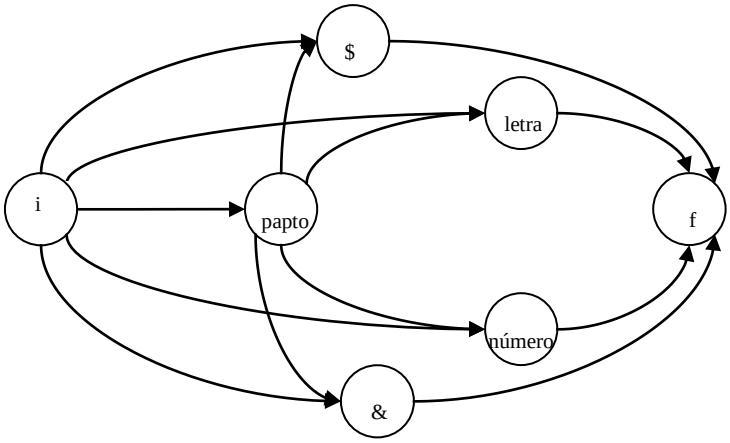
HDD internos

Escalera



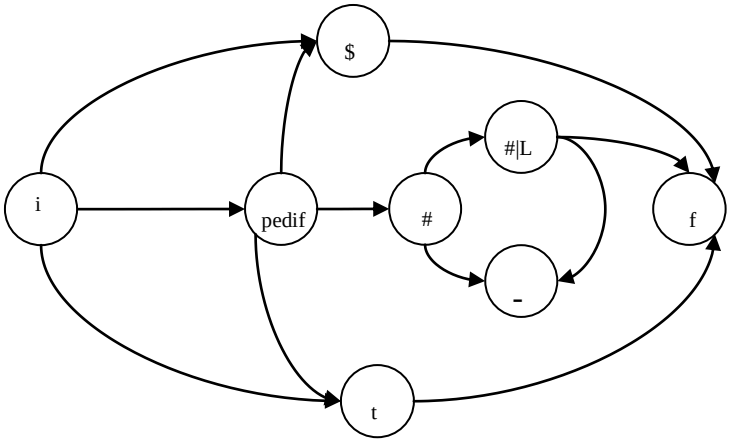
$	Número +letra
pesc	Prefijo escalera
&	Número + letra Número + número

Apartamento



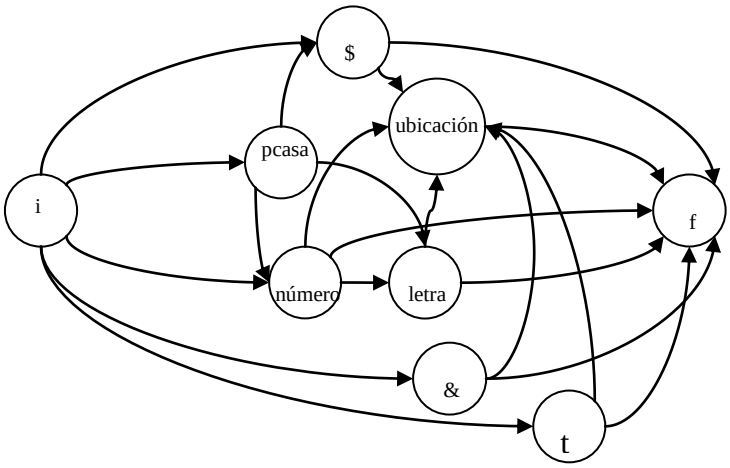
$	Número +letra
papto	Prefijo apartamento
&	Número + letra Número + número

Edificio



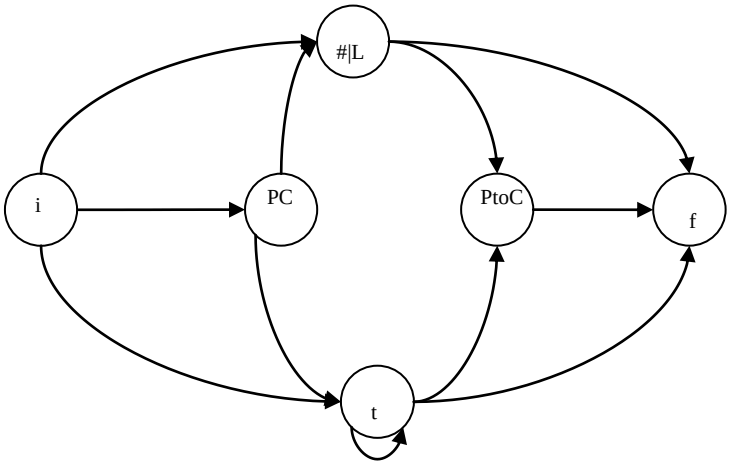
$	Número +letra
pedif	Prefijo edificio
#|L	Número entero, número ordinal o letra
#	Número
-	Carácter guión

Casa



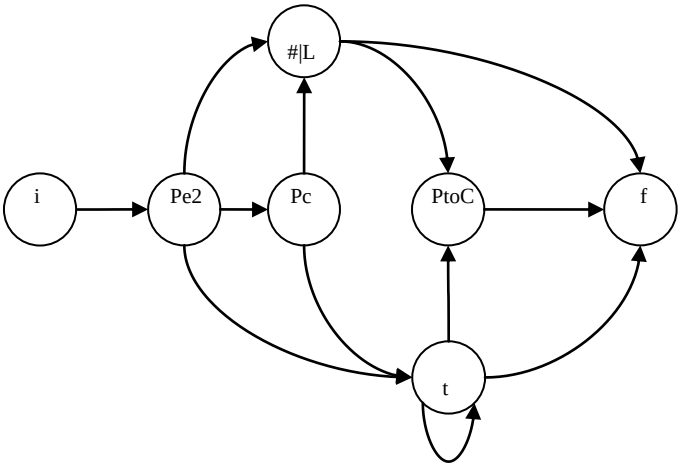
$	Número +letra
pcasa	Prefijo casa
&	Número más guión más letra, Letra más guión más número Número más guión más número
ubicación	Bajo, Alto, interior, norte, sur , este, oeste

Calle (igual a entre calle1)



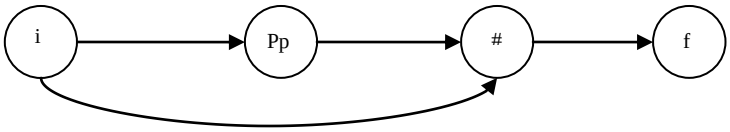
#|L	Número entero, número ordinal o letra
Pc	Prefijo calle
PtoC	Punto cardinal

Entre calle 2



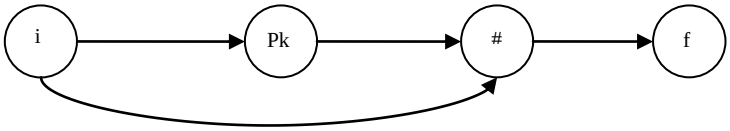
#|L	Número entero, número ordinal o letra
Pc	Prefijo calle
PtoC	Punto cardinal
Pe2	“y” ó “e”

Piso



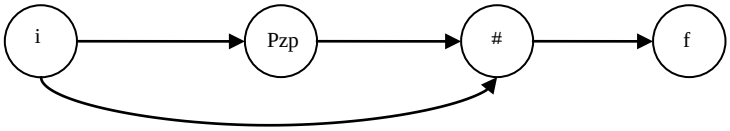
Pp	Prefijo piso
#	Número, número ordinal

Kilómetro



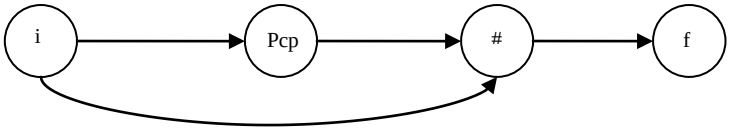
Pk	Prefijo piso
#	Número entero o entero más fracción

Zona Postal



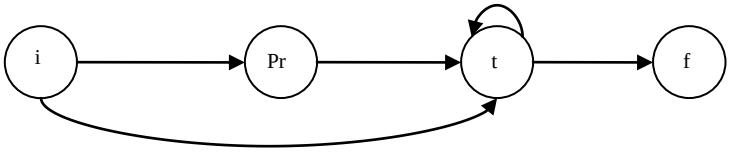
Pzp	Prefijo zona postal
#	Número

Código Postal



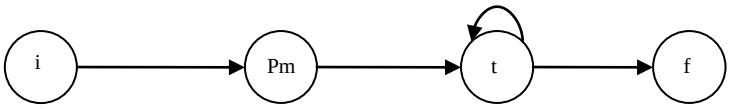
Pcp	Prefijo código postal
#	Número (5 dígitos)

Reparto



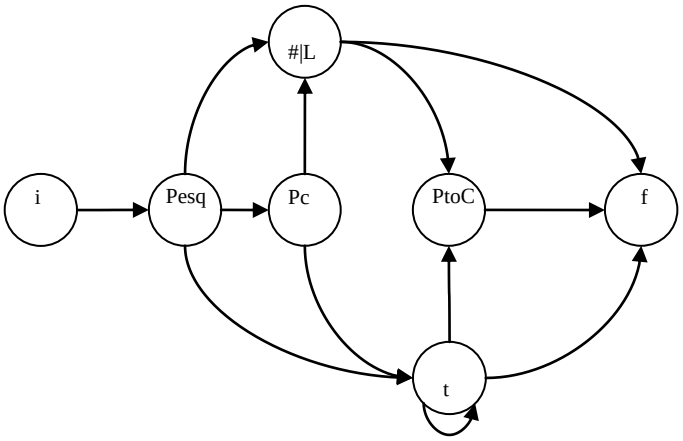
Pr	Prefijo reparto

Municipio



Pm	Prefijo municipio

Esquina



#|L	Número entero, número ordinal o letra
Pesq	Prefijo esquina
PtoC	Punto cardinal
Pc	Prefijo calle

ANEXO E

Diagrama de clases del DBAnalyzer

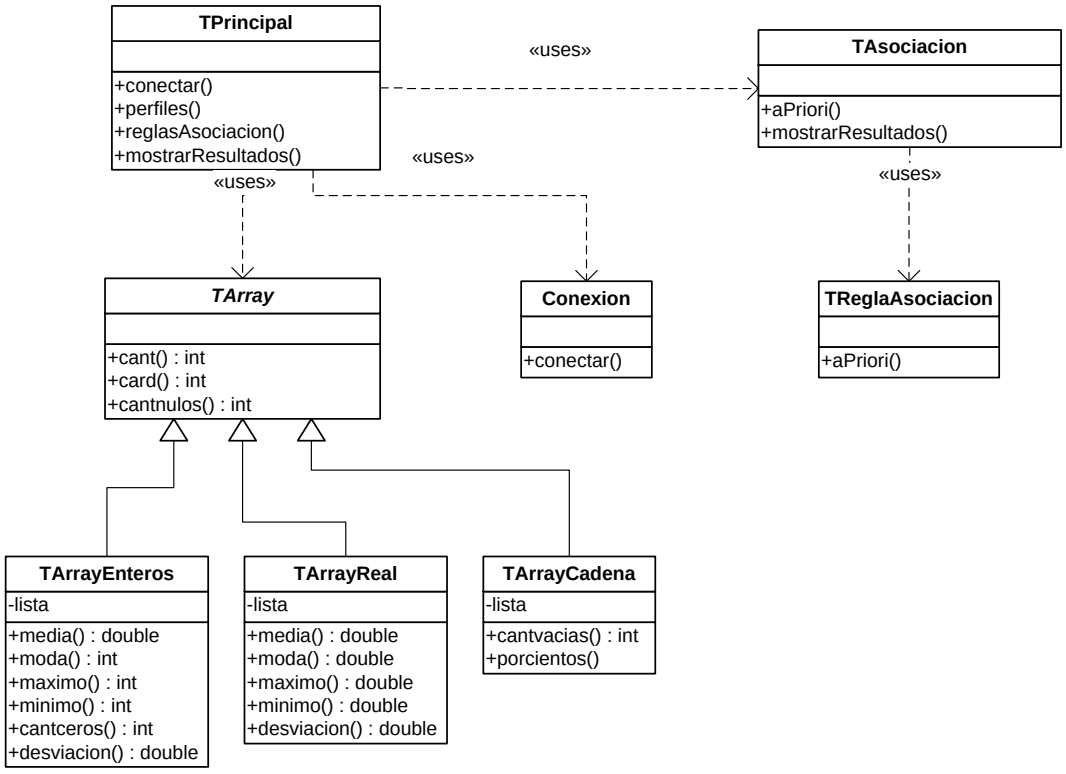
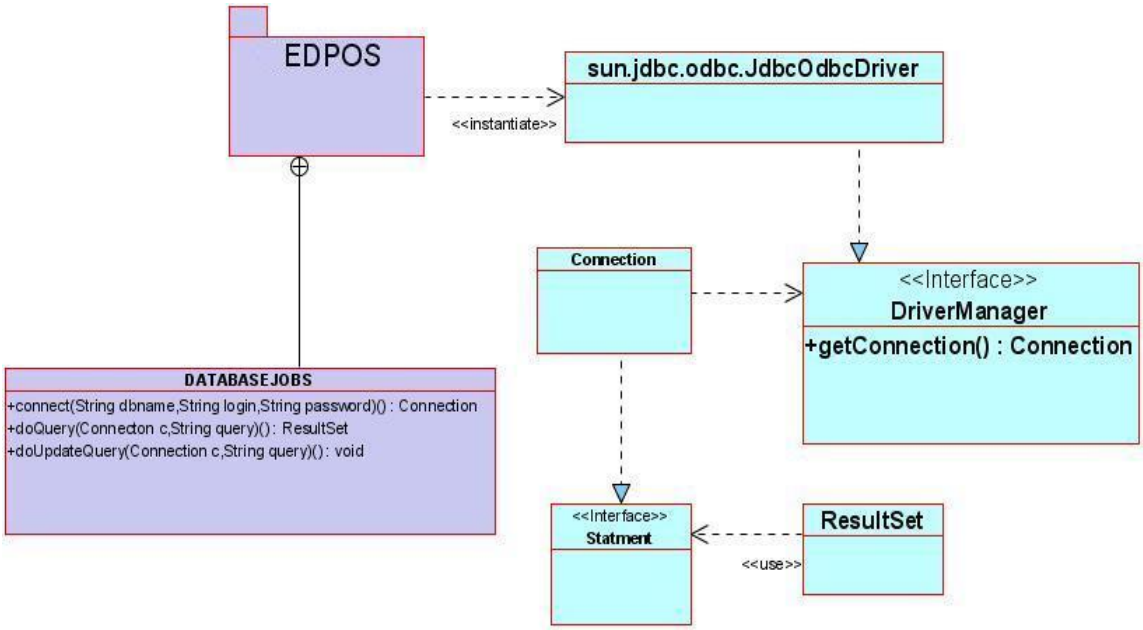


Diagrama de clases del EDPOS



ANEXO F

Cadenas agrupadas		En el orden que se introdujeron al programa. (el número del grupo no constituye un dato se escribe solo para referenciarlo)	
“ausensia"	1	"ruma"	13
"ausenvia"	1	"integrql"	10
"ausencia"	1	"ausensia"	1
"hausencia"	1	"partir"	8
"asencia"	1	"portección"	2
"aucencia"	1	"profeción"	3
		"rwma"	14
"portección"	2	"pavo"	16
"ptotección"	2	"porfesión"	3
"proteccion"	2	"mirsda"	9
"protexión"	2	"programación"	5
"proticción"	2	"asherido"	7
"protección"	2	"deshumificación"	6
		"intengral"	10
"profeción"	3	"deshumificasión"	6
"porfesión"	3	"itegral"	10
"progesión"	3	"mirasa"	9
"profesión"	3	"partor"	8
		"gracias"	12
"pergección"	4	"cavo"	15
"perfwcción"	4	"cavbo"	15
"perfección"	4	"gracia"	12
"perfeccion"	4	"rema"	14
		"psvo"	16
"programación"	5	"ausenvia"	1
"pogramación"	5	"ausencia"	1
"programacion"	5	"integral"	10
"porgramacion"	5	"hausencia"	1
"progarmacion"	5	"pogramación"	5
"progamación"	5	"programacion"	5
		"ptotección"	2
"deshumificación"	6	"rena"	14
"deshumificasión"	6	"asencia"	1
"dehumificación"	6	"porgramacion"	5
"deshumificacion"	6	"intregrar"	11
"deshumificasion"	6	"rimas"	13
"desumificación"	6	"adhetido"	7

			"proteccion"	2	
"asherido"	7		"murada"	9	
"adhetido"	7		"pabo"	16	
"adherido"	7		"protexión"	2	
"aderido"	7		"mirada"	9	
			"progesión"	3	
"partir"	8		"pergección"	4	
"partor"	8		"adherido"	7	
"pargir"	8		"dehumificación"	6	
			"profesión"	3	
"mirsda"	9		"hintegrar"	11	
"mirasa"	9		"deshumificacion"	6	
"murada"	9		"integrar"	11	
"mirada"	9		"aucencia"	1	
			"garcia"	12	
"integrql"	10		"perfwcción"	4	
"intengral"	10		"rina"		13
"itegral"	10		"proticción"	2	
"integral"	10		"cabo"	15	
			"grzcia"	12	
"intregrar"	11		"aderido"	7	
"hintegrar"	11		"deshumificasion"	6	
"integrar"	11		"integras"	11	
"integras"	11		"desumificación"	6	
			"perfección"	4	
"gracias"	12		"perfeccion"	4	
"gracia"	12		"progarmacion"	5	
"garcia"	12		"pargir"	8	
"grzcia"	12		"progamación"	5	
			"rima"	13	
"ruma"	13		"protección"	2	
"rimas"	13		"paco"	16	
"rina"	13				
"rima"	13				
"rwma"	14				
"rema"	14				
"rena"	14				
"cavo"	15				
"cavbo"	15				
"cabo"	15				
"paco"	16				
"pavo"	16				
"psvo"	16				
"pabo"	16				

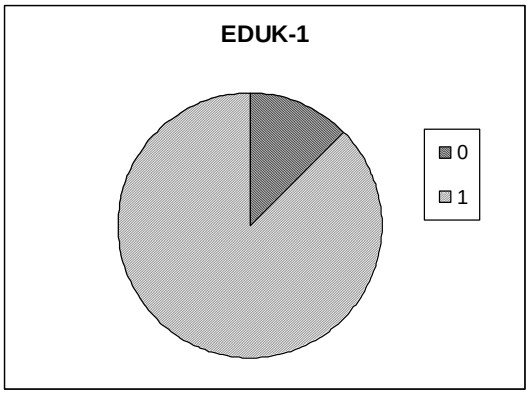
Grupos obtenidos por el programa utilizando la distancia indicada							
Distancia EDUK Costo 1		Distancia EDUK Costo 2		Distancia LD Costo 1		Distancia LD Costo 2	
partir	8	partir	8	partir	8	partir	8
partor	8	partor	8	partor	8	partor	8
pargir	8	pargir	8	pargir	8	pargir	8
programación	5	pavo	16	programación	5	pavo	16
pogramación	5	psvo	16	pogramación	5	psvo	16
programacion	5	pabo	16	progamación	5	pabo	16
porgramacion	5	paco	16			paco	16
progarmacion	5			deshumificación	6		
progamación	5	programación	5	deshumificasión	6	programación	5
		pogramación	5	dehumificación	6	pogramación	5
deshumificación	6	programacion	5	deshumificacion	6	progamación	5
deshumificasión	6	porgramacion	5	deshumificasion	6		
dehumificación	6	progarmacion	5	desumificación	6	deshumificación	6
deshumificacion	6	progamación	5			deshumificasión	6
deshumificasion	6			cavo	15	dehumificación	6
desumificación	6	deshumificación	6	cavbo	15	deshumificacion	6
		deshumificasión	6	cabo	15	deshumificasion	6
cavo	15	dehumificación	6			desumificación	6
cavbo	15	deshumificacion	6	gracias	12		
cabo	15	deshumificasion	6	gracia	12	cavo	15
		desumificación	6	garcia	12	cavbo	15
gracias	12			grzcia	12	cabo	15
gracia	12	cavo	15				
garcia	12	cavbo	15	ausensia	1	gracias	12
grzcia	12	cabo	15	ausenvia	1	gracia	12
				ausencia	1	garcia	12
rwma	14	gracias	12	hausencia	1	grzcia	12
rema	14	gracia	12	asencia	1		
rena	14	garcia	12	aucencia	1	ausensia	1
		grzcia	12			ausenvia	1
ausensia	1			integrql	10	ausencia	1
ausenvia	1	rwma	14	intengral	10	hausencia	1
ausencia	1	rema	14	itegral	10	asencia	1
hausencia	1	rena	14	integral	10	aucencia	1
asencia	1			integras	11		
aucencia	1	ausensia	1			integrql	10
		ausenvia	1	programacion	5	intengral	10
integrql	10	ausencia	1	porgramacion	5	itegral	10
intengral	10	hausencia	1	progarmacion	5	integral	10
itegral	10	asencia	1			integras	11
integral	10	aucencia	1	mirsda	9		
integras	11			mirasa	9	programacion	5

		integrql	10	murada	9	porgramacion	5
mirsda	9	intengral	10	mirada	9	progarmacion	5
mirasa	9	itegral	10				
murada	9	integral	10	asherido	7	mirsda	9
mirada	9			adhetido	7	mirasa	9
		mirsda	9	adherido	7	murada	9
asherido	7	mirasa	9	aderido	7	mirada	9
adhetido	7	murada	9				
adherido	7	mirada	9	profeción	3	asherido	7
aderido	7			porfesión	3	adhetido	7
		asherido	7	protexión	2	adherido	7
profeción	3	adhetido	7	progesión	3	aderido	7
porfesión	3	adherido	7	profesión	3		
protexión	2	aderido	7			profeción	3
progesión	3			intregrar	11	porfesión	3
profesión	3	profeción	3	hintegrar	11	protexión	2
		porfesión	3	integrar	11	progesión	3
intregrar	11	protexión	2			profesión	3
hintegrar	11	progesión	3	pergección	4		
integrar	11	profesión	3	perfwcción	4	intregrar	11
				perfección	4	hintegrar	11
pergección	4	intregrar	11	perfeccion	4	integrar	11
perfwcción	4	hintegrar	11				
perfección	4	integrar	11	ruma	13	pergección	4
perfeccion	4	integras	11	rwma	14	perfwcción	4
				rema	14	perfección	4
ruma	13	pergección	4	rena	14	perfeccion	4
rimas	13	perfwcción	4	rimas	13		
rina	13	perfección	4	rina	13	ruma	13
rima	13	perfeccion	4	rima	13	rwma	14
						rema	14
portección	2	ruma	13	portección	2	rena	14
ptotección	2	rimas	13	ptotección	2	rimas	13
proteccion	2	rina	13	proteccion	2	rina	13
proticción	2	rima	13	proticción	2	rima	13
protección	2			protección	2		
		portección	2			portección	2
saño	16	ptotección	2	saño	16	ptotección	2
salo	16	proteccion	2	salo	16	proteccion	2
sspo	16	proticción	2	sspo	16	proticción	2
sapo	16	protección	2	sapo	16	protección	2

Grupos obtenidos por el programa utilizando la distancia indicada			
QGramm		Jaro	
partir	8	partir	8
partor	8	partor	8
pargir	8	pargir	8
pavo	16	pavo	16
psvo	16	psvo	16
pabo	16	pabo	16
paco	16	paco	16
programación	5	programación	5
pogramación	5	pogramación	5
progamación	5	programacion	5
		porgramacion	5
deshumificación	6	progarmacion	5
deshumificasión	6	progamación	5
dehumificación	6		
deshumificacion	6	deshumificación	6
deshumificasion	6	deshumificasión	6
desumificación	6	dehumificación	6
		deshumificacion	6
cavo	15	deshumificasion	6
cavbo	15	desumificación	6
cabo	15		
		mirsda	9
gracias	12	mirasa	9
gracia	12	mirada	9
garcia	12		
grzcia	12	cavo	15
		cavbo	15
ausensia	1	cabo	15
ausenvia	1		
ausencia	1	ruma	13
hausencia	1	rwma	14
asencia	1	rema	14
aucencia	1	rena	14
integrql	10	ausensia	1
intengral	10	ausenvia	1
itegral	10	ausencia	1
integral	10	hausencia	1
integras	11	asencia	1
		aucencia	1

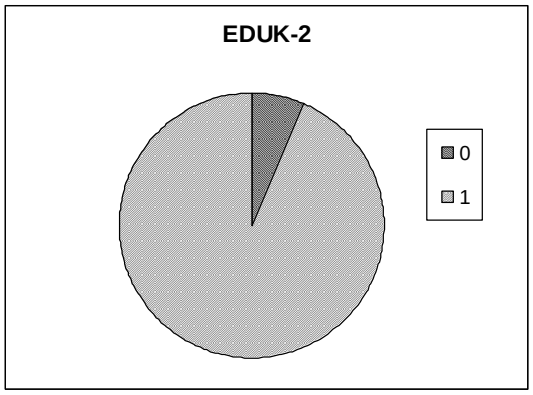
programacion	5		
porgramacion	5	integrql	10
progarmacion	5	intengral	10
		itegral	10
mirsda	9	integral	10
mirasa	9	intregrar	11
murada	9	hintegrar	11
mirada	9	integrar	11
		integras	11
asherido	7		
adhetido	7	murada	9
adherido	7		
aderido	7	asherido	7
		adhetido	7
profeción	3	adherido	7
porfesión	3	aderido	7
progesión	3		
profesión	3	profeción	3
		porfesión	3
intregrar	11	progesión	3
hintegrar	11	profesión	3
integrar	11		
		gracias	12
pergección	4	gracia	12
perfwcción	4	garcia	12
perfección	4	grzcia	12
perfeccion	4		
		pergección	4
ruma	13	perfwcción	4
rwma	14	perfección	4
rema	14	perfeccion	4
rena	14		
rimas	13	rimas	13
rina	13	rina	13
rima	13	rima	13
portección	2	portección	2
ptotección	2	ptotección	2
proteccion	2	proteccion	2
protexión	2	protexión	2
proticción	2	proticción	2
protección	2	protección	2

Tablas de frecuencias



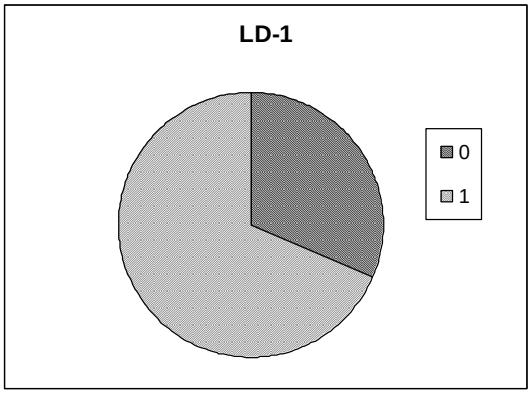
EDUK1

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	0	2	12.5	12.5	12.5
	1	14	87.5	87.5	100.0
	Total	16	100.0	100.0	



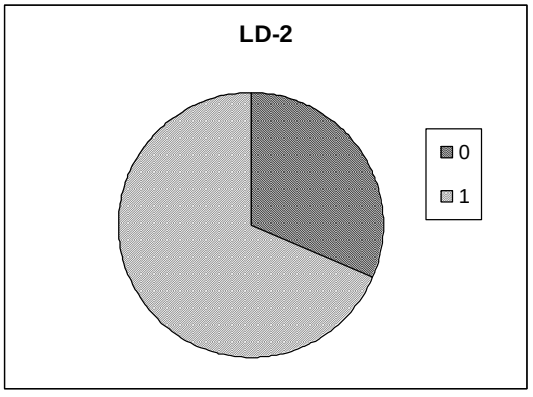
EDUK2

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	0	1	6.3	6.3	6.3
	1	15	93.8	93.8	100.0
	Total	16	100.0	100.0	



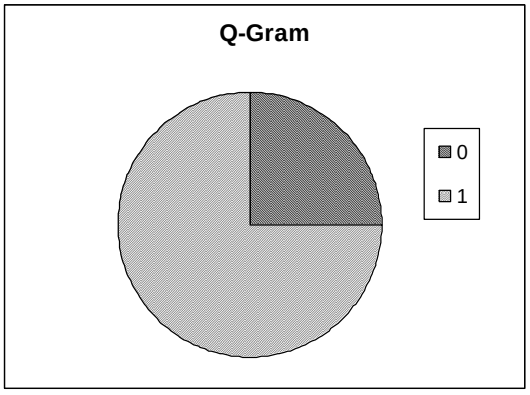
LD1

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	0	5	31.3	31.3	31.3
	1	11	68.8	68.8	100.0
	Total	16	100.0	100.0	



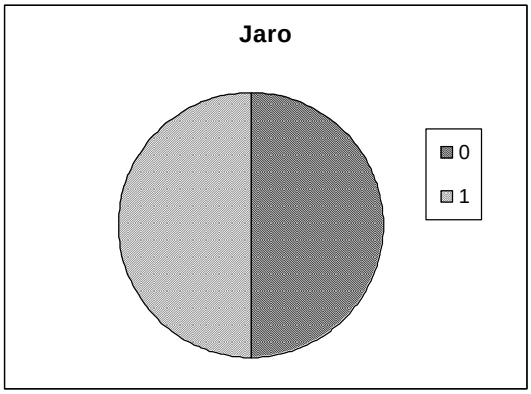
LD2

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	0	5	31.3	31.3	31.3
	1	11	68.8	68.8	100.0
	Total	16	100.0	100.0	



QGram

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	0	4	25.0	25.0	25.0
	1	12	75.0	75.0	100.0
	Total	16	100.0	100.0	



Jaro

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	0	8	50.0	50.0	50.0
	1	8	50.0	50.0	100.0
	Total	16	100.0	100.0	

Pruebas no paramétricas

Prueba de McNemar

Tablas de contingencia

Patrón y EDUK1

	EDUK1	
Patron	0	1
0	0	0
1	2	14

Patrón y EDUK2

	EDUK2	
Patron	0	1
0	0	0
1	1	15

Patrón y LD1

	LD1	
Patron	0	1
0	0	0
1	5	11

Patrón y LD2

	LD2	
Patron	0	1
0	0	0
1	5	11

Patrón y QGram

	QGram	
Patron	0	1
0	0	0
1	4	12

Patrón y Jaro

Patron	Jaro 0	Jaro 1
0	0	0
1	8	8

Estadísticos de contraste(b)

	Patron y EDUK1	Patron y EDUK2	Patron y LD1	Patron y LD2	Patron y QGram	Patron y Jaro
N	16	16	16	16	16	16
Sig. exacta (bilateral)	.500(a)	1.000(a)	.063(a)	.063(a)	.125(a)	.008(a)
Sig. exacta (unilateral)	.250	.500	.031	.031	.063	.004
Probabilidad en el punto	.250	.500	.031	.031	.063	.004

a Se ha usado la distribución binomial.
b Prueba de McNemar

ANEXO G

Medidas F-Score

El F-Score se define de la siguiente forma:

$$F = 2\left(\frac{P * R}{P + R}\right)$$

Donde:

$R = \frac{|TP|}{|TP| + |FN|}$ es denominada Recall **y**

$P = \frac{|TP|}{|TP| + |FP|}$ es denominada Precision

TP son los verdaderos positivos, (los pares de cadenas que, siendo de la misma categoría, están en el mismo grupo),

TN son los verdaderos negativos (los pares de cadenas que no pertenecen a la misma categoría y no están en el mismo grupo),

FP los falsos positivos (los pares de cadenas que, siendo de categorías diferentes, coinciden en el mismo grupo), FN los falsos negativos (los pares de cadenas que, siendo de la misma categoría, están en grupos distintos).

Dos cadenas se consideran de la misma categoría, si representan el mismo objeto.