Calidad de datos y aprendizaje automático

Detección de errores semánticos en datos estructurados con esquema desconocido

Especialización en Ciencia de Datos
Trabajo Final Integrador

Autor: Ing. Alejandro Daniel Lentini

Tutora: Lic. Valeria Inés Soliani

Noviembre de 2020

Índice

1 Introducción

La disponibilidad de sistemas informáticos trajo consigo el desarrollo de sistemas de información específicos para soportar el proceso de toma de decisiones en los distintos ámbitos de la actividad humana, siendo los datos el elemento crítico de estos sistemas: datos de una pobre calidad resultarán en decisiones deficitarias o completamente equivocadas, generando daños materiales y humanos. *Datos de calidad* son aquellos que son adecuados para el uso que se espera de ellos. De este modo, un *error*, *defecto*, *anomalía* o *problema de datos* lo constituye cualquier apartamiento de éstos respecto de sus especificaciones o requerimientos de utilización. En todo proceso de aseguramiento de la calidad de datos existe un conjunto de actividades que son usualmente referidas como *limpieza o curación de datos*, actividades cuyo propósito es detectar y corregir deficiencias en la calidad de un conjunto de datos. Suele ser frecuente que tanto el nivel de la calidad como el esquema de los datos sean desconocidos, esto es, muchos conjuntos de datos no se encuentran previamente *curados*. Es por esta razón que los usuarios deben realizar como primer paso, y muchas veces de forma manual, la inferencia del esquema y la detección de errores, realizando una primera exploración de los datos para identificar primero los tipos atómicos –números, texto, fechas, tiempo– y luego los tipos semánticos involucrados: descripciones de producto, direcciones de correo electrónico o postales, números telefónicos, números de DNI, tarjetas de crédito, y entidades como personas, instituciones, elementos geográficos –continentes, países, ciudades, compañías, etc.–. Esta es una tarea que, excepto para algunos tipos semánticos, suele ser no trivial y demanda considerable tiempo y esfuerzo, así como la intervención de expertos del dominio cuyo tiempo es preciado y escaso.

Este trabajo se encuentra organizado de la siguiente manera: en los antecedentes se introducen los conceptos y nociones esenciales necesarios para comprender la *calidad de datos*. Luego, se brinda una visión del estado del arte acerca de la *limpieza de datos*, donde se incluye una revisión de las técnicas para la *detección y corrección automática de errores*, poniendo énfasis en los *errores semánticos y sintácticos* en conjuntos de datos multivariados. Seguidamente se realiza una descripción general del problema, incluyendo a continuación la justificación, alcance y limitaciones, hipótesis y metodología de trabajo. Finalmente se presentan y comentan los resultados obtenidos en la experimentación, exponiendo las conclusiones y trabajos futuros. El trabajo termina con las referencias a la bibliografía consultada y los anexos que amplían o complementan la información expuesta en el cuerpo principal.

1.1 Antecedentes

En las sociedades actuales, prácticamente en casi todos los ámbitos se producen gran cantidad y diversidad de datos que luego son empleados para soportar, o incluso automatizar, tanto la toma de decisiones como la realización de acciones.

Los sistemas que facilitan el procesamiento y análisis de datos requieren un nivel mínimo de calidad de la información para poder producir resultados confiables y precisos. Es por esto que la cuestión de la calidad de datos ha estado presente desde los albores mismos de los sistemas de información. A esto se suma que, en la actualidad, y debido en parte a fenómenos como la cuarta revolución industrial y la digitalización, los datos son prácticamente ubicuos y la aplicación de técnicas de inteligencia artificial a los sistemas de información se encuentra en crecimiento continuo. Esto hace que la cuestión de la calidad de datos sea de total relevancia para los usuarios de los sistemas de información, tanto en los negocios y la industria, como en las organizaciones gubernamentales y hasta en la ciencia.

Como expone Sadiq [1, p. 3], en la literatura pueden encontrarse contribuciones al estudio e investigación de la calidad de datos que pueden agruparse en tres grandes perspectivas, a saber:

a. *Organizacional:* aspectos organizacionales para gestionar y asegurar la calidad de datos, definiendo objetivos y estrategias, roles, procesos, políticas y estándares.

b. *Arquitectura informática:* características y funcionalidades que deben tener los sistemas y tecnologías informáticas para garantizar la implementación de la gestión de calidad de datos definida por la perspectiva organizacional.

c. *Computacional:* algoritmos y técnicas computacionales necesarias de los sistemas y tecnologías informáticas implementadas por la perspectiva de arquitectura informática.

La investigación en temas de calidad de datos es muy profusa, y un panorama general puede encontrarse en [2]–[4], donde se analizan trabajos provenientes de las ciencias de la información y de las ciencias de la computación que proponen para la calidad de datos varios marcos de trabajo, metodologías, términos, dimensiones y taxonomías.

1.1.1 Datos, información y conocimiento

Antes de comenzar a profundizar en la calidad de datos, es necesario precisar primero la noción de *datos*, y su relación con conceptos estrechamente relacionados, pero que no deben entenderse como sinónimos: *información* y *conocimiento*. Si bien no hay definiciones únicas y taxativas en la literatura [5]–[7], hay cierto acuerdo en considerar que los *datos* son registros de la observación del mundo real –hechos–, que la *información* es el procesamiento de estos datos junto con un análisis que le otorga contexto, y finalmente el *conocimiento* sería la internalización de dicha información en la mente del usuario, cuando es entendida y sintetizada con el conocimiento previo.

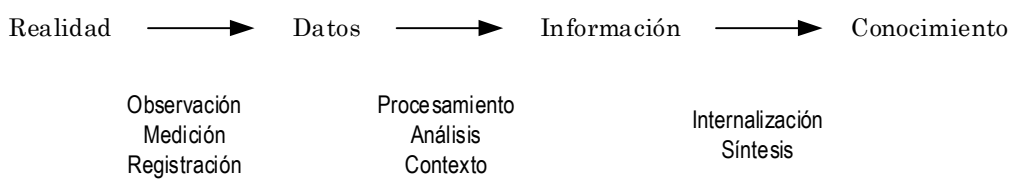


Figura 1-1. *Relación jerárquica entre los términos datos, información y conocimiento*. Los *datos* son la registración de hechos de la realidad. Adaptado de [5]–[7].

En lo que sigue se entenderá por *datos* a la conceptualización propuesta por Fox [8], donde se define *dato*[1] como el resultado de una modelización, representación y registración de hechos de la realidad, a saber:

a. *Modelización:* los hechos y eventos de la realidad que pretenden ser observados, medidos y registrados como *datos*, se modelan como una colección de duplas <entidad, {<atributo, valor>}>.

b. *Representación:* para cada dupla <atributo, valor> de una entidad, se define la regla de representación de cada atributo a, esto es, los posibles valores v pertenecientes a un dominio D, que puede tomar el atributo a.

c. *Registración:* para cada regla de representación, se define un formato de registración asociado con el medio electrónico donde los datos quedarán registrados. Por ejemplo, para el caso de atributos que representan fechas, la registración podría ser en formato americano $mm/dd/aa$, o europeo $dd/mm/aa$.

Esta definición de datos tiene la ventaja de que lleva a considerar naturalmente tres tipos de problemas de calidad de datos: i) el nivel de adecuación del modelo a la realidad, ii) la adecuación de los valores, y finalmente iii) la adecuación de la representación y registración.

1.1.2 Datos y toma de decisiones

La disponibilidad de sistemas informáticos trajo consigo el desarrollo de sistemas de información específicos para soportar el proceso de toma de decisiones en los distintos ámbitos de la actividad del hombre. Un sistema de soporte a la toma de decisiones[2] puede entenderse en términos muy generales como un sistema computacional interactivo que asiste a los usuarios en el empleo de comunicaciones, datos, documentos, conocimiento y modelos para resolver problemas y tomar decisiones [9].

Debido a la extensa gama de procesos de decisión y los diferentes ámbitos donde se aplican, existen varios tipos de sistemas de soporte a la toma de decisiones. En este trabajo serán de interés

[1] *datum* en lengua inglesa
[2] *Decision support systems*, DSS

aquellos impulsados por datos[3], categoría en la que entran los sistemas de inteligencia de negocios y sus diferentes denominaciones en la industria y los negocios (cf. [10]).

La siguiente figura conceptualiza la relación existente entre los datos, el proceso de toma de decisión y los sistemas de soporte empleados: los datos son la materia prima de los sistemas de soporte a la toma de decisión. Esta relación pone de manifiesto que los datos son el elemento crítico de este proceso: *datos de una pobre calidad resultarán en decisiones deficitarias o completamente equivocadas*, generando impactos y costos a los usuarios.

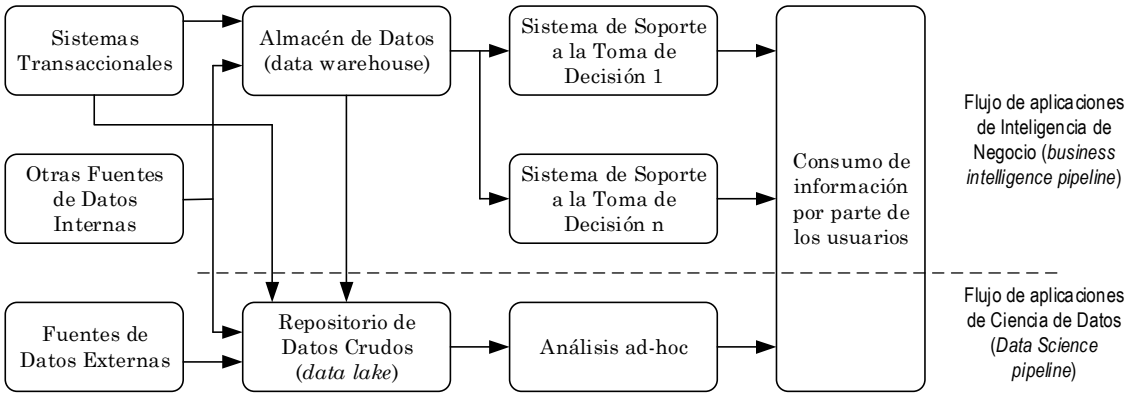


Figura 1-2. *Los datos y los sistemas de soporte a la decisión*. Es evidente que datos de una pobre calidad resultarán en decisiones deficitarias o completamente equivocadas.

1.1.3 Impactos y costos de la calidad de datos

Se ha argumentado que los sistemas que facilitan el procesamiento y análisis de datos requieren un nivel mínimo de calidad de la información para poder producir resultados confiables y precisos. Si los datos empleados para soportar decisiones son de pobre calidad, entonces necesariamente las decisiones tomadas con dichos datos serán deficitarias o completamente equivocadas, ocasionando daños materiales y humanos.

La literatura brinda un amplio análisis del impacto de la calidad de datos [11]–[17], pudiendo mencionar los siguientes como los más significativos:

a. Ineficiencias operativas.

b. Pérdida de ingresos (costos de oportunidad).

c. Insatisfacción de clientes y reclamos.

d. Demandas por incumplimiento de regulaciones.

e. Accidentes con daños materiales y pérdidas humanas.

[3] *Data-driven* DSS

Desde un punto de vista económico, estos impactos son monetizados y traducidos en *costos de la calidad de datos*, que, si bien pueden ser clasificados de diferentes maneras [18, p. 88], a grandes rasgos todas ellas proponen dos categorías principales: i) costos debido a una pobre calidad de datos, y ii) costos asociados con el mejoramiento de la calidad de los datos.

1.1.4 Calidad de datos

Si bien en la literatura no existe una definición generalmente aceptada, un criterio empleado con frecuencia para definir la *calidad de datos* es el concepto de *adecuación para el uso*. Este concepto proviene del mundo de la gestión de la calidad y fue esbozado a inicios de la segunda mitad del siglo XX por Juran [19, p. 20], el cual trasladado al ámbito de los sistemas de información permite definir como *datos de calidad* aquellos que sean adecuados para el uso que se espera de ellos. Como se aprecia, esta es una definición operacional centrada en los usuarios o consumidores de los datos, tal como expone Wang [20], y será la definición utilizada en este trabajo.

1.1.5 Dimensiones y métricas de la calidad de datos

El concepto de *adecuación para el uso* anteriormente mencionado si bien es útil es incompleto, dado que la calidad de datos es un concepto multidimensional, y para poder precisarlo en su totalidad la literatura ha desarrollado, a su vez, el concepto de *dimensiones de la calidad de datos* [8], [18, p. 19], siendo las mencionadas con mayor frecuencia la *exactitud*, *completitud*, *consistencia*, y *actualidad* de los datos.

Dimensión	Definición
Exactitud	Gado de cercanía de un valor v a algún valor v' considerado como la representación correcta (exacta) de v en la realidad. Pueden identificarse dos tipos de exactitud: *Exactitud sintáctica:* sólo interesa conocer si el valor v toma uno de los valores en el dominio definido, esto es, si $v \in D_a$. *Exactitud semántica:* interesa conocer la cercanía del valor v al valor verdadero (exacto) v', por lo que coincide completamente con el concepto de exactitud[4].
Completitud	Grado en el cual los valores de un conjunto de datos existen e incluyen los datos correspondientes a los objetos de la realidad.
Consistencia	Grado en el cual los valores de un conjunto de datos verifican reglas semánticas.
Actualidad	Grado en el cual un valor v está actualizado respecto a un marco temporal.

Tabla 1-1. *Dimensiones de calidad de datos más significativas para el caso de un único valor de un atributo*. Con las modificaciones pertinentes, estas definiciones pueden extenderse para considerar *tuplas*, *atributos* y *relaciones*. Adaptado de [18, p. 19] y [4, p. 6].

[4] En la literatura inglesa se emplean los términos *accuracy* y *correcteness*.

La tabla 1-1 proporciona las definiciones de estas dimensiones para el caso de un único *valor* de un *atributo* de una *relación*. Con las modificaciones pertinentes, estas definiciones pueden extenderse para considerar *tuplas*, *atributos* y *relaciones*.

Algunas de estas dimensiones son *intrínsecas a los datos* en el sentido que son independientes al contexto en el cual los datos son generados y usados, como por ejemplo la *exactitud,* mientras que otras dimensiones son *extrínsecas* y dependen del contexto de generación y aplicación de los datos, como la *completitud*, la *consistencia* y *actualidad*.

Como puede observarse, las dimensiones de la calidad de datos están definidas de forma cualitativa y no se hace referencia a la forma de asignar valores cuantitativos. Esto obliga a que una o más *métricas* deban ser asociadas a cada dimensión para poder cuantificarlas, por ejemplo para la exactitud semántica, una métrica puede ser la distancia entre el valor v de un atributo, y el valor v' verdadero del miso, cuando éste último sea conocido.

1.1.6 Caracterización de errores de datos

Se considerará como un *error*, *defecto*, *anomalía* o *problema de datos* a cualquier apartamiento de éstos respecto de sus especificaciones o requerimientos de utilización. Existen varias taxonomías para caracterizar los diferentes tipos de errores de datos [21]–[28], las cuales difieren en la cobertura de errores, los criterios, definiciones y terminología empleados. Una buena visión general y comparación de estas taxonomías puede encontrarse en [27]. Algunos de los criterios empleados en estas clasificaciones, en forma individual o combinada, son:

a. *Origen de datos:* una o múltiples fuentes de datos diferentes.

b. *Cantidad de relaciones (tablas/archivos):* una o varias relaciones relacionadas entre sí por un esquema.

c. *Nivel de aplicación: extensivo* que aplica a instancias de los datos, e *intensivo* que hace referencia a la estructura o esquema de los datos.

d. *Relación con el contexto o dominio de aplicación:* independencia o dependencia.

1.1.7 Marcos de trabajo y metodologías

Se han desarrollado varias metodologías para abordar el problema de la calidad de datos, algunas de aplicación general y otras específicas a determinados contextos de aplicación. Análisis comparativos de dichas metodologías pueden encontrarse en [4], [29], [30], [31, p. 353].

Como detalla Batini *et al.* [4], [31, p. 353], en el caso más genérico, la secuencia de actividades de una metodología de calidad de datos está compuesta por tres etapas, las cuales se sintetizan en la figura 1-3.

Por otro lado, Loshin [32, p. 166] propone una metodología orientada hacia la mejora continua con un círculo virtuoso compuesto de 5 etapas, mientras que fuera del ámbito académico, la asociación internacional de gestión de datos DAMA[5] propone un marco de trabajo integrador para la gestión de datos en su guía DAMA-DBOK [33]. La consideración de aspectos de seguridad en la calidad de datos es abordada por Pavón *et al.* [34], donde se propone una aproximación criptográfica para la evaluación de la calidad de datos.

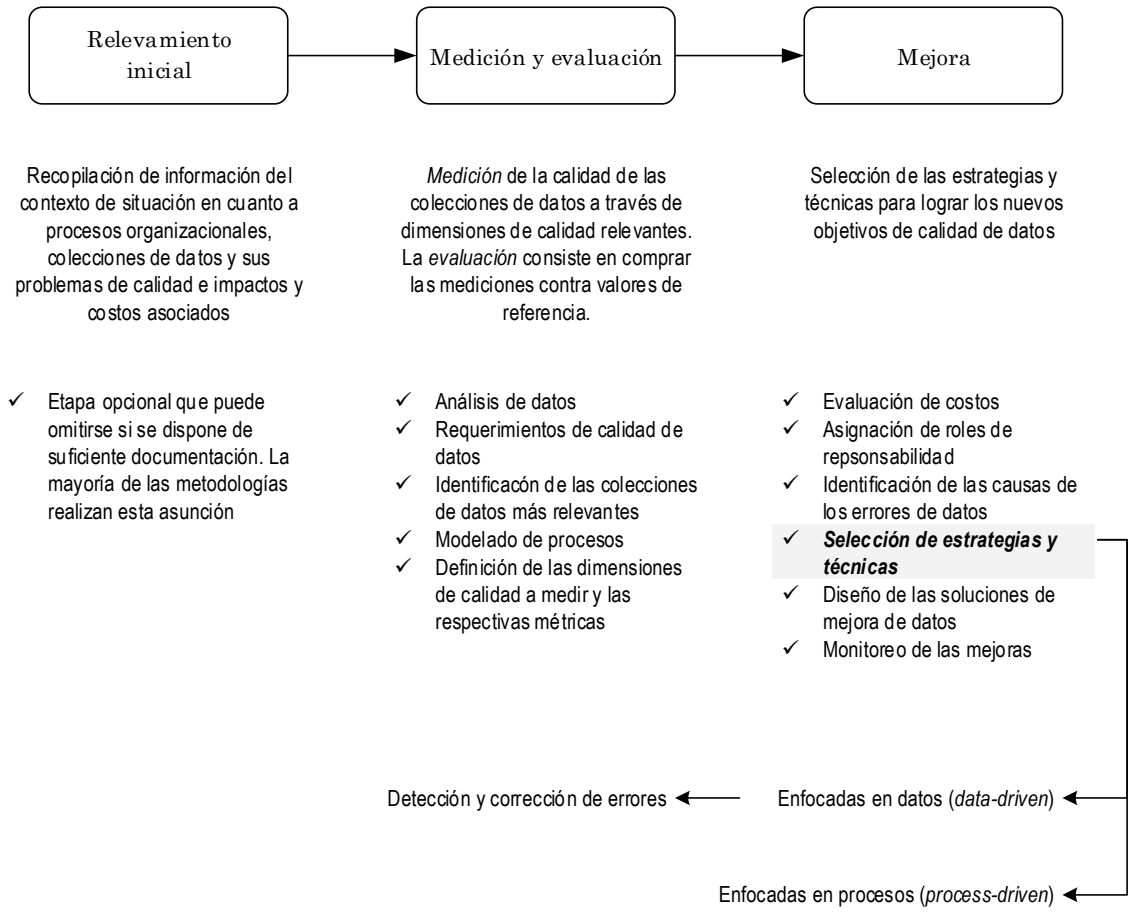


Figura 1-3. *Etapas generales de las metodologías de calidad de datos*. Se resalta la *detección y corrección de errores* que es el tema que aborda el presente trabajo. Adaptado de [4], [31, p. 353].

1.2 Limpieza de datos

En todo proceso de aseguramiento de la calidad de datos es posible encontrar un conjunto de actividades que son usualmente referidas como *limpieza de datos*[6], las cuales tienen el propósito de detectar y corregir deficiencias en la calidad de un conjunto de datos.

[5] DAMA es una institución mundial sin fines de lucro que nuclea a profesionales técnicos y de los negocios con el propósito de impulsar el desarrollo de prácticas de gestión de datos e información. https://dama.org/content/mission-vision-purpose-and-goals. Consultado el 13/04/2020.

[6] *Data cleaning* o *data cleansing*.

La limpieza de datos puede agruparse en tres etapas, a saber [35, p. 3]:

1. *Definición de metadatos*[7]*:* consiste en realizar un análisis del conjunto de datos para identificar o inferir el dominio de los atributos, relaciones entre atributos, tuplas duplicadas y otros patrones –en el modelo relacional esto es expresado generalmente como *restricciones de integridad*[8]–. La *definición de los metadatos* puede realizarse manualmente, automáticamente o con una combinación de ambas:
    a. *Especificación de metadatos manual:* es realizada por expertos del dominio y produce una especificación de la calidad de datos muy completa, incluyendo no solo errores sintácticos, sino semánticos. Desafortunadamente el tiempo de los expertos de dominio suele ser escaso y oneroso, por lo que no es una opción para cada conjunto de datos "sucio" que se pretenda limpiar.
    b. *Descubrimiento automático de metadatos:* consiste en realizar un análisis automático del conjunto de datos para detectar o inferir la especificación de la calidad de datos, la cual resulta ser de menor cobertura que la obtenida por la especificación manual por parte de los expertos de dominio.
2. *Detección de errores:* los varios errores y violaciones a la calidad de datos son identificados y posiblemente validados por usuarios.
3. *Corrección de errores:* actualizaciones a las colecciones de datos son aplicadas o sugeridas a los usuarios, de forma tal de hacer que los datos verifiquen los requerimientos de calidad.

Las técnicas de detección de errores pueden a su vez ser:

a. *Cuantitativas:* usualmente involucran métodos estadísticos para identificar valores anormales, por lo que han sido ampliamente estudiados en el contexto de la *detección de anomalías*.
b. *Cualitativas:* son enfoques descriptivos que especifican los patrones o restricciones que una instancia de datos debe verificar para ser considerada de calidad.

Otros autores proponen variaciones menores a la organización del proceso de limpieza de datos, por ejemplo, agregando una primera fase de auditoría de datos [36], o bien incorporando un último paso de control o post-proceso [37].

La siguiente figura sintetiza el proceso de limpieza de datos, el cual comienza con un conjunto de datos "sucios" a limpiar. El primer paso consiste en determinar los metadatos, lo cual puede realizarse de forma manual por expertos del dominio, de forma automática, o por una combinación

[7] En este trabajo se entiende por *metadatos* a aquellos datos que son empleados para proveer información sobre otros datos, esto es, *datos acerca de otros datos*: descripciones semánticas de los objetos de la realidad que representan, descripciones de estructura y formato de almacenamiento, referencias a otras tablas, medidas estadísticas, etc.

[8] *Integrity constraints*: conjunto de restricciones IC que en el modelo relacional un esquema de base de datos relacional $S = \{R_1, R_1, \dots, R_n\}$ debe satisfacer. Restricciones a nivel de esquema son las de dominio, valor NULO, clave, integridad de entidad e integridad referencial. Existe otro tipo de restricciones que son las *semánticas*, denominadas a veces como reglas del negocio, y cuya definición es dependiente del contexto de aplicación.

de ambas. Con la información provista por los metadatos, el siguiente paso consiste en la detección de las instancias del conjunto de datos –atributos de tuplas, atributos y tuplas enteras– que violan o tienen una alta probabilidad de no verificar la especificación de calidad. Por último, el proceso termina con la corrección de las instancias identificadas como errores. Los pasos de detección y corrección pueden realizarse también de forma manual, automática, o con una combinación de ambas.

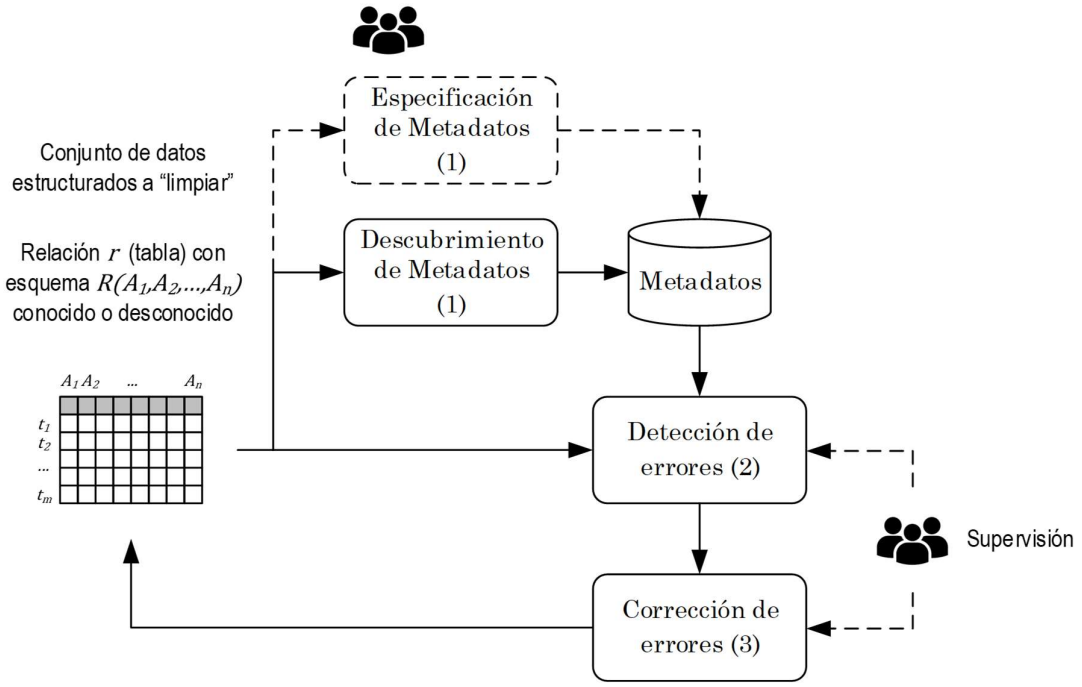


Figura 1-4. *Proceso de limpieza de datos*. A partir de datos "sucios" a limpiar, se determinan los metadatos para luego detectar y finalmente corregir las instancias con errores. Todas las etapas pueden realizarse de forma manual, automática o por una combinación de ambas. Adaptado de [35]–[37]

En las siguientes secciones se comentará la literatura relacionada con el perfilamiento de datos y la detección de errores, quedando sin cubrir la corrección de errores dado que no es objeto del presente trabajo.

1.2.1 Panorama general

Dasu y Johnson [38] abordan la limpieza de datos con la aplicación iterativa de técnicas de análisis exploratorio de datos. Visengeriyeva y Abedjan [39] proponen dos enfoques para la combinación de técnicas de detección de errores, ambos basados en el descubrimiento de metadatos y aprendizaje de modelos: el primer enfoque agrega el resultado individual de todas las técnicas de detección, mientras que el segundo filtra aquellas técnicas que son irrelevantes para el conjunto de datos a limpiar. La arquitectura propuesta emplea módulos de detección de errores y descubrimiento de metadatos que se ejecutan en paralelo, y cuyos resultados son empleados luego

por un *módulo agregador*, que es el responsable de combinar los resultados de las diferentes técnicas de detección empleando modelos basados en *redes neuronales*, *árboles de decisión* y *Bayes ingenuo*. Abedjan *et al.* [40] llevan adelante un estudio empírico para analizar los resultados de 8 herramientas en la detección de 4 tipos de errores –valores atípicos, violaciones a restricciones, violaciones a patrones y duplicados– en 5 conjuntos de datos reales. Las conclusiones principales son: i) no hay una herramienta dominante que cubra todos los conjuntos de datos y tipos de errores, ya que éstas alcanzaron en promedio 47% de *precisión*[9] y 36% de *exhaustividad*[10], lo que indica que es necesario realizar una combinación de herramientas para mejorar los resultados; y ii) el orden de aplicación de las herramientas mejora la precisión obtenida, a la vez que disminuye el costo de validación por parte de usuarios. Chu *et al.* [41] ofrecen un panorama de técnicas de detección de error cualitativas –restricciones de integridad, reglas y patrones de datos correctos– y enumeran varios desafíos a la hora de implementar la limpieza de datos. Rahm y Do [26] realizan una clasificación de errores en el contexto de los procesos de ETL de almacenes de datos, diferenciando según la cantidad de fuentes a considerar y según sea el nivel de aplicación –esquema o instancia de datos–, brindando finalmente un panorama de herramientas comerciales. Müller y Freytag [37] comparan diferentes técnicas en función del tipo de anomalías –errores léxicos, de formato, valores faltantes, irregularidades, violaciones a restricciones de integridad, así como tuplas faltantes, duplicadas o inválidas– y luego ofrecen una comparación de cinco proyectos de limpieza de datos. Maletic y Marcus [36], [42] hacen un repaso general de la limpieza de datos, presentan un marco general para finalmente comentar y testear algunos métodos: detección estadística de anomalías, reconocimiento de patrones, segmentación y técnicas de minería de datos. Hellerstein [43] ofrece una visión general de técnicas cuantitativas para limpieza de datos en grandes bases de datos, comentando técnicas paramétricas y no paramétricas basadas en estadística robusta, tanto para casos univariados como multivariados. Para escenarios donde la eficiencia es crítica menciona algunos algoritmos que computan de forma aproximada los valores requeridos en una única pasada por el conjunto de datos. Finalmente, también brinda referencias sobre métodos para datos de tipo categórico. Pearson [44] aborda el problema de los *valores nulos implícitos*[11], que a diferencia de los valores nulos explícitos –ausencia de valor–, son codificaciones que indican que realmente no existe un valor. Krishnan, Haas, Franklin *et al.* [45] realizaron una investigación acerca de cómo se llevan adelante los procesos de limpieza de datos entrevistando a analistas de datos e ingenieros de infraestructura de datos. Viendo que la limpieza de datos suele ser frecuentemente un proceso iterativo diseñado a la medida de los diferentes casos de uso, los autores proponen una serie de recomendaciones para mejorar la aplicación de los desarrollos actuales de la academia con las prácticas de la industria.

[9] *Precisión* = verdaderos positivos / (verdaderos positivos + falsos positivos).
[10] *Exhaustividad* = verdaderos positivos / (verdaderos positivos + falsos negativos).
[11] *DMV: disguised missing values*.

1.2.2 Descubrimiento de metadatos

Un panorama general de la *definición de metadatos*[12] lo ofrece Naumann [46], mientras que Abedjan. Golab y Naumann [47] realizan un tratamiento más profundo donde comentan que los metadatos pueden ir desde estadísticas simples –número de columnas y filas, información de esquema y tipos de datos, número de valores diferentes, número de valores únicos, distribuciones de frecuencia de valores, número de valores nulos, etc.– a metadatos más complejos, como información acerca de múltiples columnas y tablas y sus relaciones –*claves principales*[13]*, foráneas candidatas*[14]*, dependencias funcionales*[15] *y de otro tipo*–. Abedjan, Golab y Naumann [48], [49] realizan una revisión general del perfilamiento de datos, clasificando las tareas tradicionales por dimensionalidad –una única columna, múltiples columnas, y múltiples columnas entre diferentes tablas– para luego comentar casos de uso y posteriormente analizar las principales técnicas y herramientas por tipo de tara. Por otro lado, Yan y He [50] ofrecen un sistema para sintetizar lógicas de detección de *tipos de datos semánticos*[16], donde el usuario especifica el tipo de datos objetivo y el sistema busca en repositorios de código abierto para sintetizar las potenciales lógicas de detección. Hulsebos *et al.* [51] proponen una detección de tipos de datos semánticos con una red neuronal profunda multi entrada, la cual fue entrenada en más de 680 mil columnas de datos reales, siendo capaz de predecir 78 tipos semánticos. Zhang *et al.* [52] desarrollaron un sistema que combina *aprendizaje profundo* con *modelado de tópicos* para mejorar la detección de datos semánticos considerando la relación contextual entre los distintos atributos de un conjunto de datos, siendo capaz también de predecir 78 tipos semánticos y habiéndose entrenado con un subconjunto de los datos que emplearon Hulsebos *et al.* Otro enfoque para la detección de tipos de datos semánticos es aplicar técnicas de *procesamiento de lenguaje natural*, específicamente técnicas para el *reconocimiento de entidades nominadas*[17] [53]–[55], donde por ejemplo Krataithong, Buranarach, Hongwarittorrn y Supnithi emplean dicha técnica para inferir el tipo semántico de columnas de archivos de texto plano [56]. Por último, y como comentan entre otros Hulsebos *et al.* [51], los enfoques tradicionales para inferir tipos semánticos de datos están basados en técnicas de búsqueda empleando expresiones regulares o diccionarios, donde para esta última técnica se utilizan algoritmos de búsqueda eficientes como por ejemplo la búsqueda binaria [57], [58].

[12] *Data profiling.*
[13] *Primary key*: es la clave empleada para identificar unívocamente a las tuplas de una relación, siendo alguna de las claves candidatas de dicha relación. Una clave es un conjunto de atributos K tal que dos tuplas cualesquiera de la relación no puedan tener idénticos valores para sendos atributos, esto es, $t_i \neq t_j \; \forall \; i \neq j$. Además, debe ser una *superclave mínima*: si se remueve uno de los atributos, la restricción de unicidad ya no se verifica.
[14] *Foreign Key*: conjunto de atributos FK que referencian a las tuplas t_1 de una relación R_1 las tupla t_2 de una relación R_2
[15] *Functional dependency*: existe una dependencia funcional entre dos conjuntos de atributos X y Y de una relación R, cuando los valores de X determinan los valores de Y, esto es, $X \rightarrow Y$
[16] Los *tipos de datos semánticos* refieren a la relación de los datos con las entidades del mundo real que representan, mientras que los *tipos de datos atómicos* corresponden a los tipos primitivos empleados para representar dichas entidades. Por ejemplo, para una entidad países, un atributo que refleje el nombre de los mismos, tendrá un tipo atómico de texto, mientras que el tipo semántico sería país.
[17] *NER*: *named-entity recognition* consiste en reconocer las entidades del mundo real a la que hacen referencia ciertas palabras del lenguaje: nombres, lugares, empresas, etc.

1.2.3 Descubrimiento de restricciones

Fan, Geerts, Jia y Kementsietsidis [59] estudian la aplicación de *dependencias funcionales condicionales* para la detección de inconsistencias de datos. Chu, Ilyas y Papotti [60], [61] proponen un enfoque para la corrección de errores empleando *restricciones de negación*[18] con predicados ad-hoc –operadores para incluir valores numéricos: <, >, =, etc.– que tiene en cuenta las interacciones entre diferentes restricciones. Bleifub, Krusey Naumann [62] desarrollaron un algoritmo –Hydra– para la detección de *restricciones de negación* con un complejidad de ejecución linear en el número de tuplas. En cuanto a la complejidad computacional de los algoritmos para descubrir restricciones, Wei, Leck y Link [63] comentan que el problema de decisión asociado al descubrimiento de *restricciones de unicidad embebidas* [64, p. 9] es NP-completo y W[2]-completo en el tamaño de la entrada.

1.2.4 Detección de valores atípicos

Hawkins [65, p. 1] define un *valor atípico*, o *anomalía*, como una observación que se desvía bastante de otras observaciones, de forma que levanta sospechas de que fue generada por un mecanismo diferente. De manera similar, Barnett [66] define como observación atípica aquella que parece desviarse marcadamente de otros miembros de la muestra a la que pertenece. Una revisión de los principales métodos de detección de anomalías puede encontrarse en [35], [67], [68], mientras que un tratamiento personalizado se encuentra lo ofrecen [69], [70]. Como se señala en las referencias mencionadas, la detección de anomalías presenta al menos tres desafíos: i) *criterio o patrón de normalidad:* definición de qué es lo que se considerará normal, y qué anormal; ii) *interferencia*: las anomalías pueden afectar la implementación de ciertos patrones o criterios, como por ejemplo en las técnicas que emplean estimadores estadísticos, y en estos casos, el problema se resuelve recurriendo a estimadores robustos, ya que no se ven afectados por la presencia de valores atípicos en el conjunto de datos; iii) *dimensionalidad:* a medida que la dimensionalidad del conjunto de datos se incrementa, la dificultad algorítmico-computacional de las técnicas se incrementa también, pero su efectividad tiende a decrecer. D'Urso [71] propone un enfoque que emplea tres módulos de detección basados en *algoritmos basados en proximidad y segmentación*, los cuales son aplicados secuencialmente, de forma tal que cuando los resultados de un módulo son confirmados por otro, entonces son identificados como anomalías potenciales.

1.2.5 Minería de datos y Aprendizaje automático

Hipp *et al*. [72] proponen la aplicación de minería de datos a la calidad de datos en lo que ellos denominan *minería de calidad de datos*[19], analizando luego la técnica de *reglas de asociación*. Farzi y Dastjerdi [73] también aplican *reglas de asociación*, con la consideración de dependencias

[18] *Denial constraints (DC)* son una generalización de las restricciones de integridad (IC).
[19] *Data quality mining,* DQM

funcionales y ofreciendo una métrica para la determinación de la calidad de una transacción de una base de datos. Grüning [74] analiza el empleo de *clasificadores SVM* para identificar y corregir errores de consistencia de datos. Dai, Yoshigoe y Parsley [75] desarrollaron un enfoque basado en redes neuronales profundas junto con control estadístico para la detección de valores atípicos. Conceptualizando los errores de datos como *ruido presente en conjuntos de datos*, una revisión de la literatura es provista por Gupta y Gupta [76], especialmente para los escenarios de clasificación y predicción. Para la identificación de *ruido de clase* –errores en el atributo de una relación que es considerado como la clase a clasificar/predecir–, se comenta que las técnicas empleadas son *clasificadores simples* –árboles y redes neuronales–, *ensamblado de clasificadores* –buscando que diferentes clasificadores propongan diferentes clases y así encontrar potenciales errores–, y técnicas basadas en distancias para evaluar la cercanía entre diferentes instancias. En cuanto a la identificación de *ruido de atributos*, mencionan el algoritmo PANDA [77]. Finalmente, mencionan tres enfoques posibles para el tratamiento del ruido, a saber: a) ignorarlo, b) filtrarlo, y c) modificarlo[20]. En la misma línea que el trabajo recién descrito, Zhu y Whu [78] evalúan el impacto del ruido en el proceso de aprendizaje de modelos, midiendo el ruido de clase y de atributos en 17 conjunto de datos, proponiendo para la identificación de ruido en atributos un enfoque consistente en construir un filtro de ruido entrenando un clasificador cuya clase a predecir será justamente el valor del atributo. Para no violar las hipótesis del problema de clasificación, este método es válido para aquellos atributos que tengan una correlación significativa con otros y con la clase, debiéndose emplear para los atributos no correlacionados, otros métodos como los de segmentación o asociación. Kubica y Moore [79] proponen un enfoque para identificar *campos corruptos* empleando un *modelo probabilístico* compuesto de tres elementos: un *modelo generativo* de las instancias de datos correctas, un *modelo generativo* de los valores con ruido, y un modelo probabilístico del proceso de corrupción. La aplicación del enfoque se realizó a tres conjuntos de datos sin procesamiento previo –dos correspondientes a imágenes y uno a estadísticas de páginas web– y a dos conjuntos de datos a los que se les introdujo corrupción de forma artificial. Hu *et al.* [80] también proponen un enfoque probabilístico con *redes bayesianas* para entrenar el modelo generativo de los datos, y un enfoque de máxima entropía para modelar las fuentes de error.

1.2.6 Sistemas y herramientas

Wei y Link [81] desarrollaron DataProf, un *perfilador de datos semántico* para el descubrimiento automático de *restricciones de unicidad embebidas*, que ofrece además la generación de *muestras Armstrong* [82], que son subconjuntos de los datos a limpiar que tienen la particularidad de satisfacer las mismas restricciones que las del conjunto al que pertenecen. Papenbrock, Bergmann, Finke *et al.* [83] proponen Metanome, una herramienta para el descubrimiento de metadatos focalizado en *combinaciones únicas de columnas, dependencias de inclusión, funcionales y de orden*, que permite la incorporación de algoritmos de perfilamiento por parte del usuario. Una revisión de

[20] Métodos conocidos también como *polishing, data scrubbing* o *relabeling.*

otras herramientas de perfilamiento de datos –ProLOD++, Bellman, Potter's Whel, Civilizer, Data Auditor, RuleMiner, MADLib– puede encontrarse en [47, p. 97].

En lo que respecta a detección de errores, un análisis exhaustivo de 8 herramientas –DBoost, DC-Clean, OpenRefine, Trifacta, Pentaho, Knime, Katara y TMAR– puede encontrarse en el estudio experimental de Abedjan *et al.* [40] (ver conclusiones en sección panorama general) . Krishnan y Wu [84] desarrollaron AlphaClean, un marco de trabajo para optimizar *procesos secuenciales*[21] de limpieza de datos: los usuarios definen métricas de calidad de datos con sumas de consultas agregadas de SQL, luego el sistema genera, busca y secuencia transformaciones candidatas que maximizan las métricas indicadas por los usuarios. De, Hu, Chen *et al.* ofrecen BayesWipes [85], un método para corregir errores de atributos en conjuntos de datos estructurados empleando un modelo generativo Bayesiano. Das *et al.* [86] desarrollaron Falcon, una solución para realizar *identificación de entidades*[22] empleando *colaboración abierta distribuida*[23] y *aprendizaje activo*[24]. Yakout, Elmagarmid, Neville y Ouzzani [87] ofrecen una reparación de datos guiada, GDR, que emplea *teoría de la decisión* y *aprendizaje activo* para cuantificar los beneficios de las potenciales reparaciones, y que luego utiliza información del usuario para entrenar un modelo que eventualmente pueda reemplazar la intervención humana. Rekatsinas *et al.* [88] presentan HoloClean, un sistema para corregir errores basado en aprendizaje estadístico e inferencia probabilística. Empleando como entradas el conjunto de datos a corregir, un conjunto de *restricciones de negación*[25] y opcionalmente datos externos como diccionarios, el sistema genera *modelos gráficos probabilísticos* cuyas variables aleatorias capturan la incertidumbre de las tuplas, con resultados de *precisión* promedio del 90%, *exhaustividad* promedio del 71% y una mejora del *índice F1*[26] mayor a 2x respecto de otros métodos de vanguardia. Stonebraker, Bruckner, Ilyas *et al.* [89] desarrollaron Data Tamer, un *sistema punta a punta*[27] de curación de datos para aplicar en la integración de varios conjuntos de datos, realizando únicamente *resolución de entidades*[28] como operación de limpieza de datos. Mahdavi *et al.* [90] señalan que los principales inconvenientes a la hora de aplicar algoritmos de limpieza de datos son: i) no hay una única solución que cubra todos los casos, ii) el proceso de limpieza debe ser realizado iterativamente por los usuarios hasta lograr resultados aceptables y iii) la parametrización de las técnicas requiere de un proceso de ensayo y error por parte de los usuarios. Los autores proponen un *orquestador de flujos de trabajo de limpieza de datos* empleando *aprendizaje automático* y *descubrimiento de metadatos* que utiliza la información de procesos de limpieza realizados en el pasado para identificar y proponer las mejores técnicas para un nuevo conjunto de datos. El módulo de estimación del rendimiento –índice F1– de estrategias de detección de errores corresponde al sistema REDS[29], desarrollado por Mahdavi y Abedjan [91] que considera el concepto de *"perfiles de suciedad"*[30], los cuales hacen comparables

[21] *Pipelines*.
[22] *Entity matching.*
[23] *Crowdsourcing*.
[24] *Active learning*.
[25] En el apartado *Análisis de integridad* de esta misma sección puede encontrarse trabajos relacionados.
[26] *Índice F1* = 2 * precisión * exhaustividad / (precisión + exhaustividad)
[27] *End-to-end system*.
[28] Entity resolution.
[29] https://github.com/BigDaMa/reds
[30] Dirtiness profiles

conjuntos de datos respecto de su *"suciedad"*. Estos perfiles cubren tres dimensiones de similaridad –contenido, estructura y calidad– pudiendo opcionalmente el usuario incorporar información adicional. Los perfiles son determinados por un *perfilador de datos* [31] el cual descubre automáticamente los metadatos necesarios, mientras que la estimación del rendimiento se realiza con *modelos de regresión* previamente entrenados con resultados de análisis pasados. Madden, Abedjan Fernandez *et al.* [92] proponen RAHA, un sistema de detección de errores que no requiere configuración de parámetros por parte del usuario, ya que genera sistemáticamente un amplio rango de configuraciones de técnicas de detección que luego junto con una pequeña cantidad de supervisión –inspección y anotación[32] (correcta/limpia o incorrecta/sucia) de no más de 20 tuplas– permite entrenar clasificadores para realizar la predicción en todo el conjunto de datos. Kandel *et al.* [93] desarrollaron Profiler, una herramienta de análisis visual para la detección de errores en datos tabulares: empleando técnicas de *minería de datos* se identifican potenciales errores, para luego realizar una *recomendación automática de visualizaciones* que mejor ayudarían a analizar en contexto los potenciales errores. Krishnan, Wang, Wu *et al.* [94] presentan ActiveClean[33], una marco de trabajo para la limpieza de datos iterativa en problemas de entrenamiento de *modelos estadísticos de pérdida convergente* –regresión lineal, SVM– que ofrece propiedades de convergencia. Qahtan, Elmagarmid, Fernandez et al. [95] presentan FAHES, un sistema para la detección de *valores nulos implícitos* basándose en técnicas de detección de anomalías tanto para valores numéricos como para valores categóricos de texto.

1.3 Definición del problema

Cuando se desarrollan sistemas de soporte a la toma de decisión, como por ejemplo proyectos de inteligencia de negocio y ciencia de datos, una de las primeras etapas consiste en realizar un *entendimiento de los datos* con los que se trabajará [96], [97]. Suele ser frecuente que tanto el nivel de la calidad como el esquema de los datos sean desconocidos: muchos conjuntos de datos no se encuentran previamente *curados*[34], desconociendo tanto el tipo atómico como semántico de los mismos, y desconociendo a su vez aquellos datos que no están en conformidad con dichos tipos. Es por esta razón que los usuarios deben realizar como primer paso, y muchas veces de forma manual, la inferencia del *esquema* y la *detección de errores*, esto es, realizar una primera exploración de los datos para identificar primero los tipos atómicos –números, texto, fechas, tiempo– y luego los tipos semánticos involucrados: descripciones de producto, direcciones de correo electrónico o postales, números telefónicos, números de DNI, tarjetas de crédito, y entidades como personas, instituciones, elementos geográficos –continentes, países, ciudades, compañías, etc–. Esta es una tarea que, excepto para algunos tipos semánticos, suele ser no trivial y demanda considerable tiempo y esfuerzo, así como la intervención de expertos del dominio cuyo tiempo es preciado y escaso.

[31] Data profiler
[32] *Labeling*: anotación por parte del usuario de la clase correcta en el conjunto de datos de entrenamiento –*training set*–.
[33] https://activeclean.github.io/
[34] En este trabajo se entenderá por *datos curados* a aquellos datos que han sido analizados por humanos y que han sido transformados para garantizar que se ajustan a un determinado esquema –tipo de datos atómico y semántico– y por lo tanto verifican una calidad de datos determinada por dicho esquema.

1.4 Justificación del estudio

El presente trabajo se realiza con el propósito de evaluar la aplicabilidad de técnicas del procesamiento natural del lenguaje para inferir el tipo semántico de datos y realizar una primera detección semiautomática de errores en datos estructurados multivariados con calidad y esquema de datos desconocidos, comparando los resultados con técnicas de búsqueda en diccionarios. Los resultados obtenidos permitirán concluir si estas técnicas pueden ser utilizadas en el desarrollo de *herramientas de exploración de datos* que asistan a los usuarios en la etapa de entendimiento de datos de los proyectos de inteligencia de negocio o ciencia de datos, permitiendo reducir el tiempo humano necesario para la inferencia de esquemas y la evaluación de la calidad de datos.

1.5 Alcance y limitaciones de la investigación

1.5.1 Alcance

Los resultados y las conclusiones de esta investigación están dirigidos a los usuarios de sistemas de información –analistas de datos, científicos de datos, ingenieros de datos, etc.– que realizan procesos de limpieza de datos, centrándose específicamente en la inferencia de esquemas y detección de errores semiautomática.

1.5.2 Limitaciones

Este trabajo se limitará a:

i. Conjuntos de datos estructurados multivariados, esto es, una tabla de m filas y n columnas, o equivalentemente una relación de m tuplas y n atributos.

ii. Análisis de los *valores de los datos (nivel extensivo)*, quedando fuera de alcance el tratamiento de la *calidad del modelo de datos (esquema: nivel intensivo)*. Queda excluido el análisis de si el esquema empleado es el que mejor se ajusta al modelo de la realidad donde los datos provienen.

iii. Errores a nivel de atributos en una única relación, quedando excluido los errores en dos o más atributos de una misma relación, así como los errores en dos o más tablas relacionadas.

iv. En cuanto a errores semánticos, sólo se considerará la inferencia del tipo de datos semántico de nombres de personas, y entidades geográficas como ciudades y países. Esta elección responde a que son entidades que suelen encontrarse con frecuencia en conjuntos de datos de aplicación práctica.

v. Algoritmos y técnicas de aprendizaje automático implementadas en librerías de código abierto de Python que presenten factibilidad de aplicación en una computadora personal

de prestaciones estándares para el análisis de datos, quedando excluidas técnicas de procesamiento distribuido en la nube y técnicas que no posean librerías de código abierto de Python y que por lo tanto requieran una implementación computacional eficiente en lenguajes compilados de medio nivel como C o C++.

vi. Conjuntos de datos estructurados de origen público o generados sintéticamente que se adapten de la mejor forma a los objetivos de la experimentación.

1.6 Hipótesis

Para un conjunto de datos estructurados multivariados de esquema desconocido es posible emplear técnicas de aprendizaje automático provenientes del área del procesamiento natural del lenguaje para detectar en los atributos errores semánticos a través de la inferencia de su tipo semántico.

1.6.1 Variables

Para evaluar una técnica de detección de errores se considerará que la misma puede tratarse como si fuera un problema de clasificación binaria, donde el resultado arrojado por la técnica de detección puede clasificarse como positivo cuando coincide con la clase a predecir, o negativo cuando no. Por otro lado, es necesario tener presente que los errores, como clase a predecir, están presentes en los conjuntos de datos con una baja proporción, por lo que la clasificación es desbalanceada o desequilibrada. Con estas consideraciones las *métricas de evaluación* a ser empleadas son:

i. *Precisión (P):* cuantifica la proporción de identificaciones positivas correctas:

$$P = \frac{VP}{VP + FP}$$

donde VP y FP son los verdaderos y falsos positivos respectivamente.

ii. *Exhaustividad (R):* cuantifica la proporción de positivos reales identificados correctamente:

$$R = \frac{VP}{VP + FN}$$

donde FN representa los falsos negativos.

iii. *Índice F1:* constituye una ponderación –media armónica– de la precisión y exhaustividad según:

$$F1 = 2\frac{PR}{P + R}$$

1.7 Objetivos de la investigación

1.7.1 Objetivo general

El presente trabajo tiene como objetivo general evaluar si técnicas del aprendizaje automático provenientes del área del procesamiento natural del lenguaje pueden tener aplicación práctica en la detección semiautomática de errores semánticos en datos estructurados multivariados con calidad y esquema de datos desconocidos, ofreciendo lineamientos para el desarrollo de herramientas que asistan a los usuarios en estas tareas.

1.7.2 Objetivos específicos

I. Relevar el estado del arte para obtener un panorama general de la aplicación de técnicas de aprendizaje automático para la inferencia automática de esquemas y la posterior detección de errores.

II. Evaluar la técnica de *reconocimiento de entidades (NER)* del área del *procesamiento natural del lenguaje (NLP)* para inferir el tipo semántico de un atributo de una relación (campo o columna de una tabla).

III. Comparar los resultados con técnicas de programación tradicional (expresiones regulares, algoritmos de búsqueda, etc.) analizando ventajas y desventajas en cuanto a precisión, rendimiento computacional y esfuerzo requerido para la implementación de la técnica.

IV. Elaborar lineamientos de arquitectura para el desarrollo de herramientas que asistan a los usuarios en la inferencia de esquemas y la detección de errores.

1.8 Metodología

1.8.1 Arquitectura

La siguiente figura muestra la arquitectura que se empleará en el presente trabajo. Se partirá de un conjunto de datos estructurados multivariados a limpiar (una tabla de m filas y n columnas), el cual se encontrará almacenado en un archivo de texto plano DSV[35]. Un primer *módulo perfilador* será el encardado de realizar un descubrimiento de los metadatos del conjunto D: cantidad de filas y columnas, tipo básico o atómico de datos –numérico (entero y decimal), alfanumérico, alfabético, fecha y hora, booleano–, valores nulos explícitos, valores únicos, distribución de frecuencias. Luego, el *módulo detector de errores* emplea los metadatos para detectar distintos tipos de errores. Tanto

[35] *Delimited-separated values*, archivos para registrar información tabular donde los registros (filas) se corresponden con líneas de texto, y donde los campos (columnas) de un registro se separan por algún caracter especial, como coma (CSV), tabulación (TSV), etc.

para el descubrimiento de metadatos como para la detección de errores, las distintas técnicas a aplicar son tomadas de un conjunto de técnicas disponibles, que son las que se pretenden evaluar. Finalmente, los metadatos y los errores detectados son expuestos en una interfaz para visualización por parte del usuario.

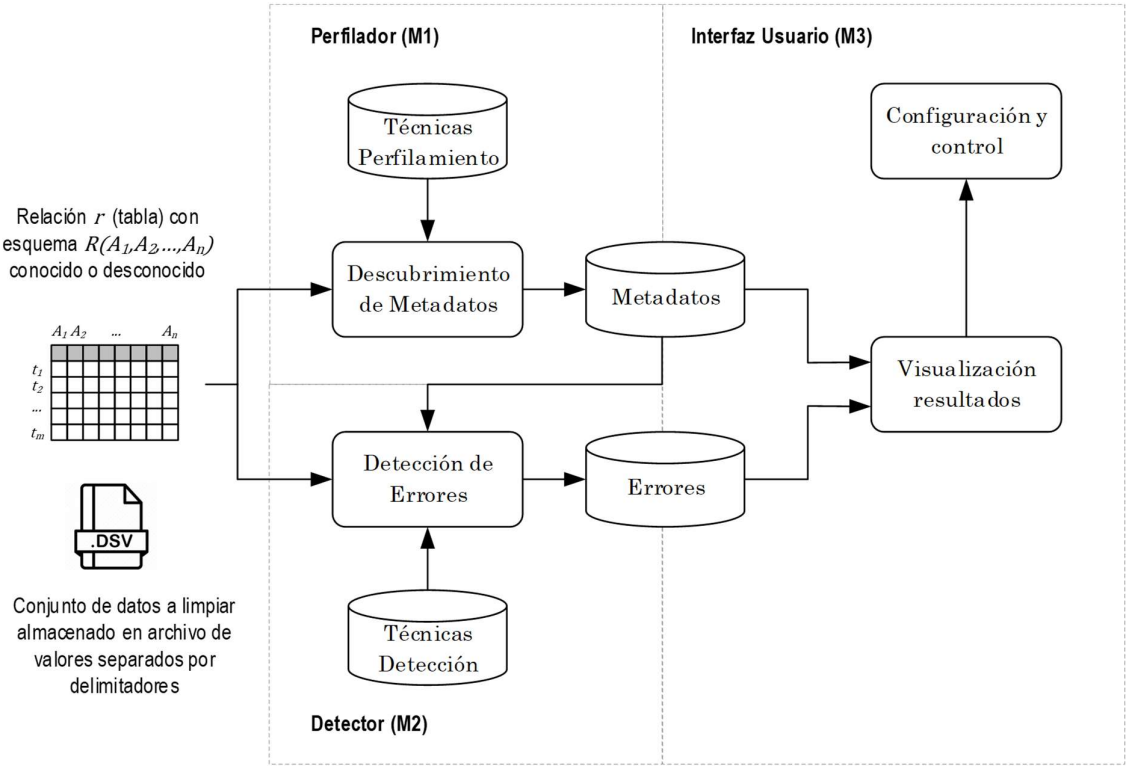


Figura 1-5. *Esquema modular de la arquitectura empleada para testear técnicas de perfilamiento de datos y detección de errores*. Partiendo de un conjunto de datos a limpiar, el *módulo perfilador* es el encargado de descubrir los metadatos, que servirán para que luego el *módulo de detección* pueda identificar potenciales errores en el conjunto de datos. Finalmente, los resultados se exponen en el módulo de la interfaz con el usuario.

1.8.2 Formalización del problema

Se asume la existencia de un conjunto de datos estructurados multivariados que posee errores que se desean detectar. Estos *datos a limpiar*, denotados por D, se considerarán como una relación r (tabla) de m tuplas (filas) $t_i =< d_{i1}, d_{i2}, \dots, d_{ij}, \dots, d_{in} >$ con esquema desconocido (columnas) $R(A_1, A_2, \dots, A_j, \dots, A_n)$, lo que implica que se desconocen tanto los dominios atómicos y semánticos, $dom_A(A_j)$ y $dom_S(A_j)$, de los atributos como las restricciones de integridad IC de dicha relación. Los valores individuales d_{ij} (celdas) corresponden al atributo j-ésimo de la tupla i-ésima (celda de la columna j-ésima y la fila i-ésima). Para la inferencia de los dominios atómicos de los atributos $dom_A(A_j)$ se aplica a cada uno de ellos un conjunto de técnicas de perfilamiento $M_A = \{m_{A1}, m_{A2}, \dots, m_{Ak}, \dots\}$ cuyo objetivo es determinar el tipo de datos atómico, esto es, $m_{Ai}[A_j] \in \{número, fecha, booleano, booleano\}$. De manera similar, para la inferencia de los dominios semánticos de los atributos $dom_S(A_j)$ se aplican técnicas $M_S = \{m_{S1}, m_{S2}, \dots, m_{Sk}, \dots\}$ cuyo propósito es determinar el tipo de datos semántico, esto es, $m_{Si}[A_j] \in$

$\{persona, país, ciudad, correo\ electrónico, \dots\}$. Con la información del dominio $dom(A_j)$ se aplica luego un conjunto de técnicas de detección $T = \{t_1, t_2, \dots, t_k, \dots\}$ con el propósito de identificar potenciales errores en los datos $e_{ij}^k = t_k(d_{ij})$, de forma tal que $t_k(d_{ij}) = 1$ en caso de que la técnica k-ésima identifique al valor d_{ij} como error, y $t_k(d_{ij}) = 0$ cuando no se identifica como error. Finalmente, los resultados se resumen en un conjunto de matrices de error $E = \{\boldsymbol{E}^1, \boldsymbol{E}^2, \dots, \boldsymbol{E}^k, \dots\}$ donde $\boldsymbol{E}^k{}_{mxn} = [e_{ij}^k]$.

1.8.3 Inferencia de tipo semántico de datos

El procesamiento natural del lenguaje –*NLP: natural language processing*– en términos amplios es un campo de estudio interdisciplinario entre la lingüística y las ciencias de la computación que investiga como las computadoras interaccionan con el lenguaje humano natural, siendo algunas de las tareas más representativas del NLP el *procesamiento de texto y audio*, el *análisis morfológico, sintáctico y semántico*. Dentro del *análisis semántico*, se encuentra la *semántica léxica* que trata de decodificar el significado de los símbolos lingüísticos dentro de un contexto dado, por ejemplo, el significado de las palabras en una oración. Una técnica popular de semántica léxica es el *reconocimiento de entidades –NER: named entity recognition–* que consiste en identificar elementos atómicos en un texto para clasificarlos en categorías semánticas –*named entities*, entidades nombradas– predefinidas como por ejemplo personas, organizaciones, localizaciones, expresiones de tiempo, cantidades, valores monetarios, porcentajes, etc. Generalmente un sistema NER recibe como entrada un *texto no anotado* y devuelve como salida un *texto anotado* donde se han identificado las *entidades semánticas*. La siguiente figura ejemplifica lo descrito.

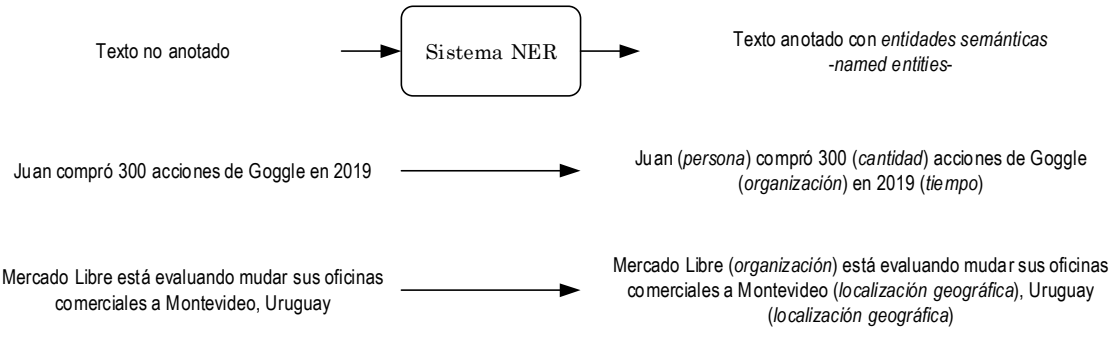


Figura 1-6. *Ejemplificación de un sistema de reconocimiento de entidades NER*. El sistema recibe como entrada un texto no anotado y devuelve como salida un texto anotado donde se han identificado las entidades semánticas.

Se empleará la técnica NER recién descrita para inferir el o los tipos semánticos de un atributo de una relación (campo o columna de una tabla). Esto es, sea m_{S1} la técnica de perfilamiento de metadatos NER a aplicar a aquellos atributos A_j de la relación r con $dom_A(A_j) \in \{alfabético\ o\ alfanumérico\}$, de forma tal que la aplicación de dicha técnica resultará en una

entidad nominada que indicará el tipo semántico de datos del atributo, esto es, $dom_S(A_j) = m_{S1}[A_j] \in \{persona, organización, localización, cantidad, tiempo, \dots\}$. Para los casos donde el atributo A_j presente más de un tipo de dato semántico, o presente instancias donde no hubo identificación de tipo, esto será considerado como una indicación de potenciales errores bajo el supuesto de que el dominio semántico $dom_S(A_j)$ debería ser único, exponiendo estos datos al usuario para su validación. La siguiente figura esquematiza la aplicación de la técnica NER para la inferencia del tipo semántico de datos y la posterior detección de errores.

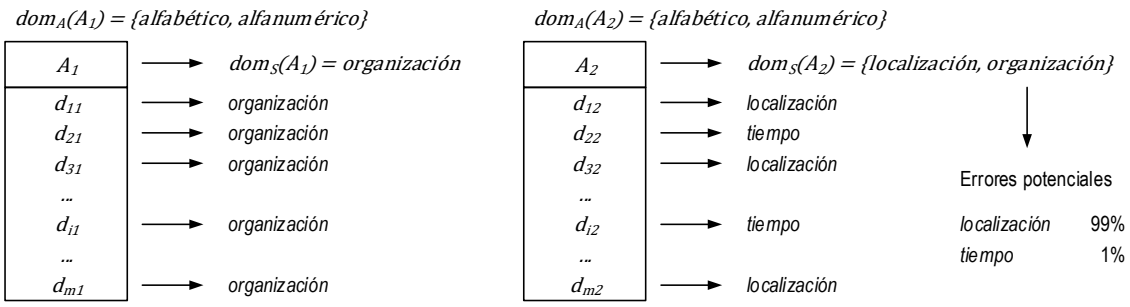


Figura 1-7. *Aplicación de la técnica NER para la inferencia del tipo semántico de datos y la posterior detección de errores*. Para aquellos atributos A_j con tipo atómico de datos alfabético o alfanumérico se aplica la técnica NER para inferir el tipo semántico a través de la *entidad nominada*. Atributos que presenten más de un tipo de dato semántico, o presenten instancias d_{ij} donde no hubo identificación de tipo, esto será considerado como una indicación de potenciales errores.

1.8.4 Experimentación

Se realizará una experimentación para determinar el rendimiento de la técnica NER en la inferencia de los siguientes tipos semánticos de datos: nombres de personas y ubicaciones geográficas como ciudades y países. La siguiente tabla resume la experimentación que se realizará en el presente trabajo: técnicas e implementaciones, tipo de error a detectar, conjuntos de datos y tipos semánticos.

Tipo de errores	Técnica - Implementación	Conjuntos de Datos	
Detección de tipo de datos atómico y nulos explícitos	Programación tradicional – Pandas	Todos	
Detección de tipo de datos semántico: *- Nombres de personas* *- Ubicaciones geográficas*	Reconocimiento de entidades – spaCy y NLTK	Names_2010Census athlete_events	world-cities worldcities

Tabla 1-2. *Características de la experimentación*: técnicas de detección que serán evaluadas en el presente trabajo, indicando la implementación utilizada, el tipo de error que detectan y los datos empleados.

En el contexto de este trabajo, se considera como *técnicas de programación tradicional* a los algoritmos que partiendo de una entrada y de una lógica definida y codificada por humanos obtienen la salida deseada, por ejemplo, buscar y ordenar valores en un conjunto, encontrar patrones en cadenas de caracteres con expresiones regulares, una función criptográfica hash, etc. Esta definición busca poner en contraste las *técnicas de aprendizaje automático*, donde no hay codificación de lógica por parte de humanos más allá de un algoritmo que *aprende* un determinado modelo matemático para poder predecir la salida en función de los datos de entrada.

Librería	Descripción general	Referencia
pandas	Extensión de la librería de cómputo científico NumPy que permite la manipulación y análisis de datos proporcionando estructuras de datos flexibles que permiten trabajar de forma eficiente con datos tabulares multidimensionales.	https://pandas.pydata.org/
sPacy	Procesamiento avanzado de lenguaje natural en Python y Cython basado en las últimas investigaciones y diseñada para ser utilizado en productos reales. Incluye modelos estadísticos pre entrenados y vectores de palabras, y actualmente admite tokenización para más de 49 idiomas. Cuenta con modelos de redes neuronales convolucionales de alta tecnología para el etiquetado, el análisis y el reconocimiento de entidades nombradas, así como una integración de aprendizaje profundo.	https://spacy.io/
NLTK	El kit de herramientas de lenguaje natural es un conjunto de bibliotecas y programas para el procesamiento del lenguaje natural (PLN) simbólico para el lenguaje de programación Python. Está destinado a apoyar la investigación y la enseñanza en procesamiento de lenguaje natural (PLN) o áreas muy relacionadas, que incluyen la lingüística empírica, las ciencias cognitivas, la inteligencia artificial, la recuperación de información, y el aprendizaje automático.	https://www.nltk.org/

Tabla 1-3. Librerías de Python utilizadas como implementación de las técnicas de perfilamiento de metadatos empleadas en el presente trabajo.

En cuanto a las implementaciones de software de las técnicas, la tabla 1-3 lista las empleadas en el presente trabajo, todas ellas librerías de código abierto de Python con gran reconocimiento y amplio uso.

1.8.5 Evaluación

Como se indicó en la sección 1.6.1, la evaluación de las técnicas de detección T se realizará considerando la precisión P, la exhaustividad R y el índice $F1$ obtenidos en una familia de conjuntos de datos de prueba $D_P = \{D_{P1}, D_{P2}, \dots, D_{Pl}, \dots\}$, de tal forma que la 3-tupla $(P_{kl}, R_{kl}, F1_{kl})$ indica la precisión, exhaustividad e índice $F1$ de la k-ésima técnica evaluada en el l-ésimo conjunto de datos.

Si los conjuntos de datos solo presentan instancias positivas de la clase a predecir, entonces corresponderá emplear únicamente la métrica exhaustividad R.

Como punto de referencia para la comparación del rendimiento de las técnicas evaluadas, tanto a nivel de capacidad de predicción como de tiempos de ejecución, se empleará la técnica de programación tradicional de *búsqueda binaria*, un algoritmo de búsqueda que encuentra la posición de un valor en un vector ordenado: se compara el valor a buscar con el elemento en el medio del vector, si no son iguales, la mitad en la cual el valor no puede estar es eliminada y la búsqueda continua en la mitad restante hasta que el valor se encuentre. En el peor de los casos se requerirá de un tiempo logarítmico, realizando $O(logn)$ comparaciones, donde n es el número de elementos del vector mientras que se requiere solamente $O(1)$ en espacio, es decir que el espacio requerido por el algoritmo es el mismo para cualquier cantidad de elementos en el vector.

1.8.6 Conjuntos de datos

Como se comentó previamente en la sección limitaciones, en este trabajo se emplearán conjuntos de datos públicos, los cuales se listan en la siguiente tabla junto con sus metadatos básicos. En el anexo I se incluyen las referencias a los mismos así como una visualización de los primeros registros con la herramienta de exploración desarrollada especialmente en este trabajo.

Nombre	Descripción general	Entidades Semánticas	Filas	Columnas
Names_2010Census	Apellidos más frecuentes según el censo de USA del año 2010	Nombres de personas	162,253	11
athlete_events	Eventos por atleta de los Juegos Olímpicos desde Atenas 1896 a Río 2016.	Ubicaciones geográficas	271,116	15
world-cities	Listado de las principales ciudades del mundo		23,018	4
worldcities	Detalle de 13,504 ciudades a finales de 2019		13,504	11

Tabla 1-4. Conjuntos de datos empleados para la evaluación de las diferentes técnicas de detección. Se incluyen los metadatos básicos.

1.8.7 Exploración de datos y visualización de resultados

Se ha mencionado que uno de los objetivos del presente trabajo es elaborar lineamientos de arquitectura para el desarrollo de herramientas que asistan a los usuarios en la exploración de datos, específicamente en las tareas de inferencia de esquemas y la detección de errores relacionados. Con este fin en mente, se desarrolló un primer prototipo de esta herramienta, la cual permite realizar una exploración interactiva de los datos, ofreciendo además una visualización de los resultados arrojados por las técnicas testeadas en la experimentación. La siguiente imagen muestra una captura de pantalla donde se pueden apreciar los metadatos del conjunto de datos, los

metadatos principales de las columnas junto con la indicación de errores, así como mayores detalles de los metadatos de una columna en particular.

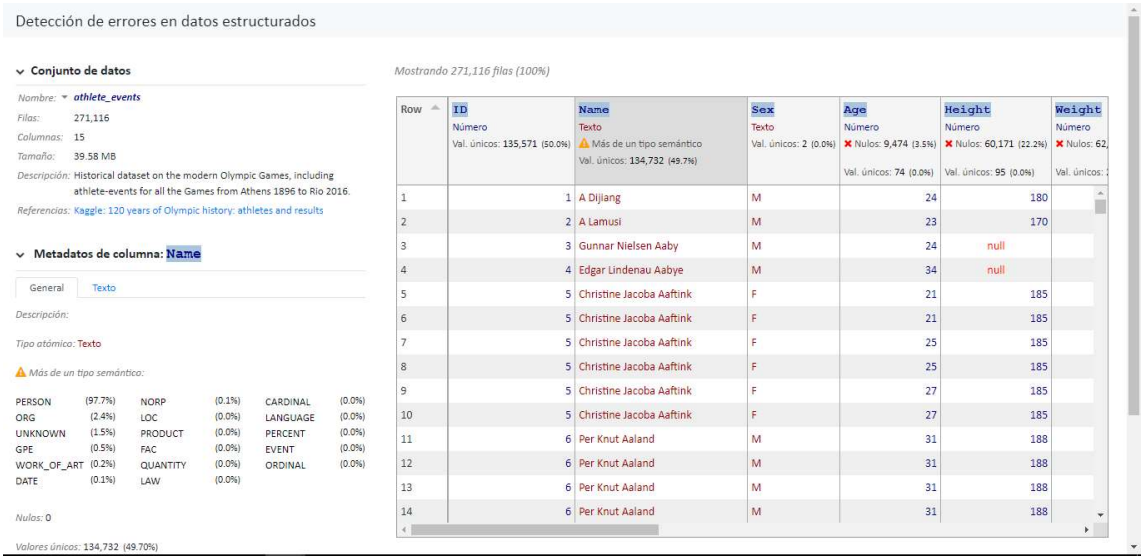


Figura 1-8. Captura de pantalla que muestra la herramienta desarrollada para explorar interactivamente los datos y visualizar los resultados de obtenidos por la experimentación. Se puede observar los metadatos del conjunto de datos, los metadatos principales de las columnas junto con la indicación de errores, así como mayores detalles de los metadatos de una columna en particular.

Esta aplicación si bien se encuentra actualmente en *versión alfa*[36], ofrece como principales funcionalidades[37]:

i. *Visualización de los metadatos de un conjunto de datos:* nombre, descripción, cantidad de filas y columnas, tamaño en disco y referencia al origen de los datos.

ii. *Visualización de los datos en formato tabular:* con la indicación del tipo atómico de datos por color, así como la indicación por columnas de la cantidad de valores nulos y diferentes advertencias, como por ejemplo la existencia de más de un tipo de datos atómico y valores nulos.

iii. *Visualización de los metadatos de una columna:* descripción, tipos atómicos y semánticos, nulos y valores únicos. Para los valores texto se listan las frecuencias, mientras que para los valores numéricos se muestran las principales medidas descriptivas –mínimo, máximo, rango, media, desvío estándar, cuartiles–, así como gráficos –histogramas y boxplots–.

iv. *Filtrado de valores:* tipo atómico y semántico, valores nulos y valores únicos.

[36] *Versión alfa*: primera versión de una aplicación de software que posee las principales funcionalidades que permiten satisfacer la mayoría de los requisitos, pero aún es un producto inestable que posee errores y donde está pendiente completar toda la funcionalidad requerida.
[37] Consultar el anexo II para obtener mayores detalles de la herramienta.

1.8.8 Software y Hardware

Intel Core i5-7300U 2.60 GHz, 2 cores, 4 threads

16.0 GB RAM, 2 x 8.0 GB SODIMM DDR4 2133 MHz

Microsoft Windows 10 Enterprise, 64-bit OS, Version 10, OS Build 16299.1806

Python 3.8.0, Numpy 1.17.4, Pandas 0.25.3, spacy 2.2.4, spacy-lookups-data 0.3.2, en-core-web-lg 2.2.5, en-core-web-sm 2.2.5, es-core-news-sm 2.2.5, nltk 3.5

2 Resultados

Se evaluó la técnica de *reconocimiento de entidades (NER)* del procesamiento natural del lenguaje (NLP) para inferir el tipo semántico de *personas* según nombres y apellidos, y *localizaciones geográficas* según ciudades, provincias –estados, departamentos, regiones, etc. según el país– y países. Como en los conjuntos de datos empleados la totalidad de las instancias d_{ij} son positivas, sólo se evaluó con la métrica *exhaustividad* (R), la cual cuantifica la proporción de positivos reales que fueron identificados correctamente como tales. A continuación, se resumirán los resultados obtenidos, pudiéndose consultar mayores detalles de forma interactiva con la herramienta de exploración de datos y visualización de resultados que se realizó para este trabajo y la cual se encuentra descrita en el anexo II.

Los resultados obtenidos muestran que la exhaustividad obtenida por esta técnica es baja y no puede ser empleada para inferir correctamente tipos semánticos de personas y localizaciones geográficas. La proporción de verdaderos positivos fue de alrededor del 43% del total de instancias –142,702 de 335,532–, resultando entre el 45% y 48% para la entidad de personas, y de aproximadamente el 20% para la entidad de localizaciones geográficas.

Tipo Semántico	Casos	sPacy				NLTK			
		VP	FN	R [%]	Tiempo Prom. [ms]	VP	FN	R [%]	Tiempo Prom. [ms]
Total	**335,532**	**142,702**	**192,830**	**42.5**	**9.233**	**144,223**	**191,309**	**43.0**	**8.829**
Personas	296,986	135,041	161,945	45.5	9.587	144,138	152,848	48.5	9.117
Apellidos	162,254	5,677	156,577	3.5	8.343	23,723	138,531	14.6	9.494
Nombres y Apellidos	134,732	129,364	5,368	96.0	11.086	120,415	14,317	89.4	8.664
Localizaciones Geográficas	38,546	7,661	30,885	19.9	6.095	85	38,461	0.2	6.123
Ciudades	35486	6,644	28,842	18.7	6.531	-	35,486	-	6.422
Países	467	408	59	87.4	6.807	85	382	34.8	17.430
Provincias	2593	609	1,984	23.5	6.062	-	2,593	-	7.288

Tabla 2-1. *Resultados generales de la aplicación de la técnica NER para inferir el tipo semántico de persona y localizaciones geográficas*.

2.1 Inferencia de la entidad personas

La inferencia del tipo de datos semántico correspondiente a *personas* arrojó una exhaustividad del 45% –135,041 de 296,986 casos–, siendo el mejor valor obtenido el correspondiente a la librería NLTK con 48.5%, mientras que la librería sPacy arrojó un resultado algo menor de 45.5%.

Se observa que la técnica NER produce una exhaustividad de al menos 90% para inferir que un determinado valor corresponde a un tipo semántico de *persona* cuando dicho valor corresponde a

nombres y apellidos –89.4% para NLTK y 96.0% para sPacy–, mientras que la exhaustividad cae significativamente cuando se trata de inferir únicamente a partir de *apellidos* –14.6% para NLTK y 3.5% para sPacy–. Estos resultados muestran que la exhaustividad es mejor cuando se dispone de varias palabras –nombres y apellidos– en lugar de tan solo una –apellido–. Como prácticamente cualquier palabra puede ser empleada como apellidos de personas, como lo demuestran los casos listados en la tabla 2-3[38], es extremadamente difícil detectar correctamente que una palabra, o conjunto de palabras, es efectivamente un apellido sin considerar otros metadatos que den mayor contexto como, por ejemplo, el nombre del atributo o atributos adyacentes. Otro factor que dificulta una correcta inferencia es que las implementaciones testeadas tienen modelos de idioma inglés, por lo que todos aquellos apellidos de lenguas diferentes a la inglesa tendrán mayores chances de ser incorrectamente procesados.

Personas	Casos	sPacy				NLTK			
		VP	FN	R [%]	Tiempo Prom. [ms]	VP	FN	R [%]	Tiempo Prom. [ms]
Total	**296,986**	**135,041**	**161,945**	**45.5**	**9.587**	**144,138**	**152,848**	**48.5**	**9.117**
Apellidos	162,254	5,677	156,577	3.5	8.343	23,723	138,531	14.6	9.494
Nombres y Apellidos	134,732	129,364	5,368	96.0	11.086	120,415	14,317	89.4	8.664

Tabla 2-2. *Resultados de la aplicación de la técnica NER para inferir el tipo semántico persona*. Se obtuvo una exhaustividad entre el 45% y 48%, la cual resulta de al menos 90% cuando se infiere a partir de *nombres y apellidos*, mientras que cae a valores entre 3 y 14% cuando se trata de inferir sólo a partir de apellidos.

Entidad Predicha	sPacy		
	Casos		Ejemplos
Total	**162,254**		
Personas	5,677	3.5%	AARON, ABASS, ABBAS, ABBY, ABDALLAH, ABDEL, ABDIKADIR, ABDOU
Organizaciones	77,394	47.7%	AABERG, AABY, AAFEDT, AAGARD, AAGESEN, AAKER, AAKHUS, AALAND
Trabajos de arte	1,550	1.0%	MARTES, BAYA, ABBASZADEH, ABDALLA, ABDUL, ABDULKAREEM, ABDULLA
Productos	172	0.1%	AULDS, BAKICH, BALANDA, BALICH, BALUSEK, BAMBURY, BANDUCCI, BAQUE
Locaciones geográficas	58	0.0%	AFRICA, ANDES, ARAGON, BABIY, BAECKER, BELLMORE, BERGLAND, BLOOR
Fechas	16	0.0%	MONDAY, FRIDAY, SATURDAY, SUNDAY, JANUARY, APRIL, JUNE, JULY, AUGUST
Nacionalidades, grupos religiosos y	14	0.0%	CHICANO, CUBAN, DUTCH, FRENCH, HARKRADER, ISRAELI, JEWS, SEDERQUIST
Cardinal	9	0.0%	BARNABEI, BILLON, HALF, NINE, SEVEN, SHETRON, SIX, TEN, THOUSAND
Ordinal	2	0.0%	FIRST, THIRD
Eventos	1	0.0%	HUGO
Sin identificar	77,361	47.7%	LUN, JUE, MON, TUE, SAT, SABADO, DOMINGO, MARZO, ABRIL, MAYO, JUNIO

[38] Se puede realizar una exploración interactiva de los resultados arrojados empleando la herramienta desarrollada a tal efecto, la cual se comenta en la sección 1.8.7.

Tabla 2-3. *Algunos resultados de la aplicación de la técnica NER para inferir el tipo semántico persona a través del apellido*. Se observa que los apellidos pueden arrojar diferentes tipos de entidades, lo cual hace que la aplicación de esta técnica no sea recomendable si es que no se considera otros metadatos como por ejemplo el nombre del atributo (título de columna) o información de atributos adyacentes.

2.2 Inferencia de la entidad localizaciones geográficas

Apenas un 20% de las instancias fueron correctamente inferidas con el tipo de datos semántico correspondiente a *localizaciones geográficas* –7,661 de 38,546 casos–, siendo sPacy la única librería capaz de detectar este tipo de entidades nombradas, ya que la librería NLTK sólo fue capaz de inferir correctamente el tipo en 85 casos –0.2%–.

Localizaciones Geográficas	Casos	sPacy				NLTK			
		VP	FN	R [%]	Tiempo Prom. [ms]	VP	FN	R [%]	Tiempo Prom. [ms]
Total	**38,546**	**7,661**	**30,885**	**19.9**	**6.095**	**85**	**38,461**	**0.2**	**6.123**
Ciudades	35,486	6,644	28,842	18.7	6.531	-	35,486	-	6.422
Países	467	408	59	87.4	6.807	85	382	34.8	17.430
Provincias	2,593	609	1,984	23.5	6.062	-	2,593	-	7.288

Tabla 2-4. *Resultados de la aplicación de la técnica NER para inferir el tipo semántico persona*. Se obtuvo una exhaustividad entre el 45% y 48%, la cual resulta de al menos 90% cuando se infiere a partir de *nombres y apellidos*, mientras que cae a valores entre 3 y 14% cuando se trata de inferir sólo a partir de apellidos.

Observando la tabla 2-4 se aprecia que el tipo de entidad mejor detectado por la técnica NER resultó ser el *país*, con una exhaustividad de 87%, continuando luego con provincias y ciudades, con proporciones de verdaderos positivos de 24% y 19% respectivamente. Estos resultados muestran nuevamente la dificultad para inferir estos tipos de entidades semánticas a partir de valores que bien pueden ser considerados como personas, organizaciones u otras entidades. Para estos casos de localización geográfica es evidente que la lengua con la que esté configurada la técnica es muy relevante, y como ya se comentó, por defecto la lengua es la inglesa, por lo que para ciudades, países y provincias con valores completamente dependientes de la lengua la clasificación será incorrecta.

Las tablas 2-5 a 2-7 muestran ejemplos de algunos resultados arrojados por la aplicación de la técnica NER para inferir la entidad locación geográfica a partir de ciudades, países y provincias. Se puede realizar una exploración interactiva de todos los resultados arrojados empleando la herramienta desarrollada a tal efecto, la cual se comenta en la sección 1.8.7.

Entidad Predicha	sPacy		
	Casos		Ejemplos
Total	**467**		
Locaciones geográficas	408	87.4%	Andorra, United Arab Emirates, Afghanistan, Antigua and Barbuda, Belgium, Greenland
Personas	25	5.4%	Saint Barthelemy, Cook Islands, Jordan, Kiribati, Saint Kitts and Nevis, Saint Lucia
Nacionalidades, grupos religiosos y	9	1.9%	American Samoa, Central African Republic, French Guiana, Palestinian Territory
Organizaciones	2	0.4%	Anguilla, Curazao
Sin identificar	23	4.9%	Bermuda, Dominica, Guadeloupe, Isle of Man, Mali, Martinique, Maldives, Niger, Mali

Tabla 2-5. *Algunos resultados de la aplicación de la técnica NER para inferir el tipo semántico localización geográfica a través del país*. Puede observarse que el 87% de los casos fueron correctamente inferidos, mientras que los casos incorrectos fueron inferidos, entre otros, como nombres de personas (5%) y nacionalidades y grupos religiosos (2%).

Entidad Predicha	sPacy		
	Casos		Ejemplos
Total	**2,593**		
Locaciones geográficas	609	23.5%	Dubai, Ajman, Kabul, Helmand, Badakhshan, Syunik Province, Ararat Province, Zaire
Personas	544	21.0%	Andorra la Vella, Umm al Qaywayn, Rio Negro, Al Fujayrah, Abu Dhabi, Sar-e Pol, Berat
Organizaciones	455	17.5%	Malanje, Namibe, Santa Fe, Corrientes, Neuquen, Tucumán, Tierra del Fuego, La Pampa
Trabajos de arte	16	0.6%	Moxico, Makamba, Bolívar, Loja, Oromiya, Imereti, Guria, Jutiapa, Tehrán, Batman
Nacionalidades, grupos religiosos y	7	0.3%	Australian Capital Territory, South Moravian, Basque Country, Tamil Nadu, Oriental
Productos	1	0.0%	Kursk
Sin identificar	961	37.1%	Herat, Kunduz, Balkh, Khowst, Badghis, Lowgar, Konar, Panjshir, Bengo, Cabinda

Tabla 2-6. *Algunos resultados de la aplicación de la técnica NER para inferir el tipo semántico localización geográfica a través de la provincia.* Tan solo el 24% de los casos fueron correctamente inferidos, mientras que los casos incorrectos fueron inferidos, entre otros, como nombres de personas (21%) y organizaciones (18%).

Entidad Predicha	sPacy		
	Casos		Ejemplos
Total	**35,486**		
Locaciones geográficas	6,644	18.7%	Barcelona, London, Antwerpen, Paris, Calgary, Albertville, Los Angeles, Salt Lake City
Organizaciones	7,124	20.1%	Torino, Grenoble, Sapporo, Garmisch-Partenkirchen, Katowice, Ibadan, Sapporo, Patna
Personas	6,675	18.8%	Rio de Janeiro, Cortina d'Ampezzo, Sankt Moritz, Sao Paulo, Rio de Janeiro, Kano
Trabajos de arte	386	1.1%	Hechi, Huaiyin, Cilacap, Shangrao, Dnipro, Nangandao, Caerdydd, Warri, Ostrava
Instalaciones	44	0.1%	La Grange Park, View Park-Windsor Hills, Great Neck Plaza, Port Loko, Port Orange
Cardinal	27	0.1%	Half Way Tree, Four Corners, Seven Corners, Five Corners, Five Forks, Three Lakes
Nacionalidades, grupos religiosos y	25	0.1%	Spanish Town, Brits, Hawaiian Gardens, Ordino, Spanish Fork, German Flatts
Productos	8	0.0%	Kursk, Salto del Guairá, Crown Point, Clarion, Rossosha, Kursk, Crown Point, Atlantis
Fechas	4	0.0%	August, Spring Lake, Winters, March
Eventos	2	0.0%	Rancho Santa Margarita
Sin identificar	14,547	41.0%	Osogbo, Indio, Ipatinga, Szczecin, Chattanooga, Kitwe, Mbale, Jiangjiafan, Piura, Ndola

Tabla 2-7. *Algunos resultados de la aplicación de la técnica NER para inferir el tipo semántico localización geográfica a través de la ciudad*. Se observar el menor valor de exhaustividad obtenido, 19% de los casos fueron correctamente inferidos. Los casos incorrectos fueron inferidos, entre otros, como organizaciones (20%) y nombres de personas (19%).

2.3 Inferencia por búsqueda binaria

Es posible inferir cualquier tipo de datos semántico con una exhaustividad del 100% empleando búsqueda binaria, siempre que se disponga de una lista con todos los valores que se pretendan inferir. Pruebas con valores de texto en una lista de más de diez millones de registros –10,207,305– arrojaron un tiempo promedio de búsqueda de 1.7µs –los detalles de la determinación del tiempo promedio de la búsqueda binaria se incluyen en el anexo III–. Este tiempo resultó ser inferior al mejor tiempo promedio obtenido por la técnica NER en más de 5,000 veces –8.8ms para la librería NLTK–.

3 Conclusiones y trabajos futuros

La aplicación de *técnicas de reconocimiento de entidades (NER)* provenientes del procesamiento del lenguaje natural (NLP) para la detección o inferencia de tipos semánticos de personas y localizaciones geográficas no produjo valores de exhaustividad aceptables que permitiesen recomendar el empleo de las mismas en herramientas prácticas. La proporción de casos verdaderos positivos fue de alrededor del 43% del total de instancias –142,702 de 335,532–, resultando entre el 45% y 48% para la entidad de *personas*, y de aproximadamente el 20% para la entidad de *localizaciones geográficas*. Los factores que explicarían los resultados obtenidos son:

i. Dado que es frecuente que una misma palabra pueda tener más de un tipo semántico, para una inferencia correcta es necesario contar con información extra que ofrezca contexto. Es por este motivo que las implementaciones actuales de las técnicas NER funcionan con valores $F1$ superiores al 90% dado que la inferencia del tipo semántico de una entidad se realiza en el contexto de una oración, no únicamente con una palabra aislada. En la aplicación de esta técnica en conjuntos de datos relacionales, la información de contexto sería considerar además del atributo que se está infiriendo, atributos adyacentes junto con los nombres de los mismos.

ii. Las implementaciones testeadas tienen modelos de idioma inglés, por lo que todos aquellos valores provenientes de lenguas diferentes a la inglesa tendrán mayores chances de ser incorrectamente procesados.

iii. Cuando consideramos la inferencia del tipo semántico de *personas*, o el de *ciudades*, es realmente una tarea muy compleja predecir matemáticamente el tipo semántico en función de alguna característica de la misma, ya que como lo muestran los conjuntos de datos empleados, así como el conocimiento popular, prácticamente cualquier palabra puede ser empleada como un nombre o apellido de personas o un nombre de ciudad.

Por otro lado, al comparar los resultados arrojados por la técnica NER con los obtenidos por una técnica de programación tradicional como es la búsqueda binaria, se observa que ésta última permitiría obtener una exhaustividad del 100% con tiempos promedio de búsqueda más de 5,000 veces mejores –1.7µs vs. 8.8ms–. Esta comparación pone de manifiesto que, en lugar de invertir una gran cantidad de esfuerzo de personal altamente calificado para construir conjuntos de datos de entrenamiento con miles de *características*[39] para que los modelos matemáticos puedan aprender los casos no detectados y así lograr mejorar la exhaustividad, puede invertirse un esfuerzo menor en construir las listas de los tipos semánticos más frecuentes –nombres y apellidos, ubicaciones geográficas, empresas, modelos de automotor, listado de productos, etc.– realizando una inferencia del tipo semántico con 100% de precisión empleando una búsqueda binaria, la cual puede ser implementada en aplicaciones de producción sin mayores consideraciones computacionales que las

[39] Una característica –*feature*– en el contexto del aprendizaje automático, es un atributo computado sobre los demás atributos de una instancia de datos. Por ejemplo, para el caso del valor de una columna, el modelo empleado en el trabajo Sherlock [51] computa un vector de 1,587 características para alimentar luego una red neuronal profunda y finalmente predecir el tipo semántico.

usualmente requeridas. Lo anterior permitiría recomendar que la detección de las entidades personas y localizaciones geográficas podría llevarse a cabo con total precisión al emplear la técnica de búsqueda binaria en diccionarios de entidades. Como mostraron los resultados obtenidos en cuanto a tamaño del diccionario y tiempos de búsqueda, podría armarse un único diccionario que englobe todos los valores a detectar y que devuelva los diferentes tipos de entidades semánticas asociadas.

Las líneas de trabajo que se continuarán desarrollando en un futuro son las siguientes:

1. Detección de nulos implícitos –aquellos valores no disponibles que se encuentran codificados con algún valor como NA, 9999, -1, etc.–
2. Detección de anomalías en atributos de tipo atómico numérico.
3. Detección de dependencias funcionales junto con las instancias que las violan.
4. Analizar el desempeño de la detección de tipos semánticos con redes neuronales profundas, similares a los trabajos Sherlock [51] y Sato [52], cuando los mismos estén disponibles como librerías de código abierto.

4 Referencias

[1] S. Sadiq, Ed., *Handbook of Data Quality*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[2] S. Sadiq, N. K. Yeganeh, and M. Indulska, "20 Years of Data Quality Research: Themes, Trends and Synergies," *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 115, pp. 153–162, 2011, doi: 10.5555/2460396.2460415.

[3] S. Sadiq, N. Khodabandehloo Yeganeh, and M. Indulska, "Cross-disciplinary collaborations in data quality research," in *19th European Conference on Information Systems, ECIS 2011*, 2011.

[4] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–52, Jul. 2009, doi: 10.1145/1541880.1541883.

[5] J. Rowley, "The wisdom hierarchy: Representations of the DIKW hierarchy," *J. Inf. Sci.*, vol. 33, no. 2, pp. 163–180, 2007, doi: 10.1177/0165551506070706.

[6] M. Frické, "The knowledge pyramid: A critique of the DIKW hierarchy," *J. Inf. Sci.*, vol. 35, no. 2, pp. 131–142, 2009, doi: 10.1177/0165551508094050.

[7] R. C. Hicks, R. Dattero, and S. D. Galup, "The five-tier knowledge management hierarchy," *J. Knowl. Manag.*, vol. 10, no. 1, pp. 19–31, 2006, doi: 10.1108/13673270610650076.

[8] C. Fox, A. Levitin, and T. Redman, "The notion of data and its quality dimensions," *Inf. Process. Manag.*, vol. 30, no. 1, pp. 9–19, 1994, doi: 10.1016/0306-4573(94)90020-5.

[9] D. J. Power, "Decision Support Systems: A Historical Overview," in *Handbook on Decision Support Systems 1*, vol. 39, no. 3, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 121–140.

[10] D. J. Power, "Understanding data-driven decision support systems," *Inf. Syst. Manag.*, vol. 25, no. 2, pp. 149–154, 2008, doi: 10.1080/10580530801941124.

[11] M. Ge and M. Helfert, "Cost and Value Management for Data Quality," in *Handbook of Data Quality*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 75–92.

[12] C. W. Fisher and B. R. Kingma, "Criticality of data quality as exemplified in two disasters," *Inf. Manag.*, 2001, doi: 10.1016/S0378-7206(01)00083-0.

[13] T. C. Redman, "Impact of poor data quality on the typical enterprise," *Commun. ACM*, 1998, doi: 10.1145/269012.269025.

[14] A. Puurunen, J. Majava, and P. Kess, "Exploring incomplete information in maintenance materials inventory optimization," *Ind. Manag. Data Syst.*, 2014, doi: 10.1108/IMDS-01-2013-0025.

[15] B. T. Hazen, C. A. Boone, J. D. Ezell, and L. A. Jones-Farmer, "Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications," *Int. J. Prod. Econ.*, 2014, doi: 10.1016/j.ijpe.2014.04.018.

[16] A. Haug and J. S. Arlbjørn, "Barriers to master data quality," *Journal of Enterprise Information Management*. 2011, doi: 10.1108/17410391111122862.

[17] S. Raghunathan, "Impact of information quality and decision-maker quality on decision quality: A theoretical model and simulation analysis," *Decis. Support Syst.*, 1999, doi: 10.1016/S0167-9236(99)00060-3.

[18] C. Batini and M. Scannapieco, *Data Quality*. Springer Berlin Heidelberg, 2006.

[19] J. Juran and A. Godfrey, *Juran's Quality Handbook*. McGraw-Hill, 1998.

[20] R. Y. Wang, "Beyond accuracy: What data quality means to data consumers," *J. Manag. Inf. Syst.*, 1996, doi: 10.1080/07421222.1996.11518099.

[21] P. Oliveira, F. Rodrigues, and H. Galhardas, "A Taxonomy of Data Quality Problems," *2nd Int. Work. Data Inf. Qual.*, no. April 2014, pp. 219–233, 2005.

[22] P. Oliveira, F. Rodrigues, and P. Henriques, "A formal definition of data quality problems," *Proc. 2005 Int. Conf. Inf. Qual. ICIQ 2005*, no. June 2014, 2005.

[23] J. M. B. Josko, M. K. Oikawa, and J. E. Ferreira, "A Formal Taxonomy to Improve Data Defect Description," in *Database Systems for Advanced Applications*, vol. 9645, no. August 2018, H. Gao, J. Kim, and Y. Sakurai, Eds. Cham: Springer International Publishing, 2016, pp. 307–320.

[24] W. Kim, B. J. Choi, E. K. Hong, S. K. Kim, and D. Lee, "A Taxonomy of Dirty Data," *Data Min. Knowl. Discov.*, 2003, doi: 10.1023/A:1021564703268.

[25] J. Schmid, "The main steps to data quality," *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.*, vol. 3275, pp. 69–77, 2004, doi: 10.1007/978-3-540-30185-1_8.

[26] E. Rahm and H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000, doi: 10.1145/1317331.1317341.

[27] N. Laranjeiro, S. N. Soydemir, and J. Bernardino, "A Survey on Data Quality: Classifying Poor Data," *Proc. - 2015 IEEE 21st Pacific Rim Int. Symp. Dependable Comput. PRDC 2015*, no. November, pp. 179–188, 2016, doi: 10.1109/PRDC.2015.41.

[28] T. Gschwandtner, J. Gärtner, W. Aigner, and S. Miksch, "A Taxonomy of Dirty Time-Oriented Data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7465 LNCS, no. January, 2012, pp. 58–72.

[29] R. Silvola, J. Harkonen, O. Vilppola, H. Kropsu-Vehkapera, and H. Haapasalo, "Data quality assessment and improvement," *Int. J. Bus. Inf. Syst.*, vol. 22, no. 1, pp. 62–81, 2016, doi: 10.1504/IJBIS.2016.075718.

[30] M. J. Eppler and D. Wittig, "Conceptualizing Information Quality : A Review of Information Quality Frameworks from the Last Ten Years," *Proc. 2000 Confernece Inf. Qual.*, no. April 2014, pp. 1–14, 2000.

[31] C. Batini and M. Scannapieco, *Data and Information Quality*. Springer International Publishing, 2016.

[32] D. Loshin, "Data Quality," in *Business Intelligence*, Elsevier, 2013, pp. 165–187.

[33] DAMA International, *DAMA-DMBOK: Data Management Body of Knowledge (2nd Edition)*. 2017.

[34] J. Y. Pavón, R. S. Lima, and H. Dí. Pando, "Evaluation of data quality: A cryptographic approach," *Comput. y Sist.*, vol. 23, no. 2, pp. 557–568, 2019, doi: 10.13053/CyS-23-2-2899.

[35] I. F. Ilyas and X. Chu, *Data Cleaning*. 2019.

[36] J. Maletic and A. Marcus, "Data Cleansing: Beyond Integrity Analysis.," *Iq*, pp. 1–10, 2000.

[37] H. Müller and J. Freytag, "Problems, Methods, and Challenges in Comprehensive Data Cleansing," *Challenges*, no. HUB-IB-164, pp. 1–24, 2003.

[38] T. Dasu and T. Johnson, *Exploratory Data Mining and Data Cleaning*. 2003.

[39] L. Visengeriyeva and Z. Abedjan, "Metadata-Driven Error Detection," *SSDBM '18 Proc. 30th Int. Conf. Sci. Stat. Database Manag.*, pp. 1–12, 2018, doi: https://doi.org/10.1145/3221269.3223028.

[40] Z. Abedjan *et al.*, "Detecting Data Errors : Where are we and what needs to be done ?," *Proc. VLDB Endow.*, pp. 993–1004, 2016, doi: https://doi.org/10.14778/2994509.2994518.

[41] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, "Data cleaning: Overview and emerging challenges," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2016, doi: 10.1145/2882903.2912574.

[42] J. I. Maletic and A. Marcus, "Data Cleansing: A Prelude to Knowledge Discovery," in *Data Mining and Knowledge Discovery Handbook*, Boston, MA: Springer US, 2009, pp. 19–32.

[43] J. M. Hellerstein, "Quantitative Data Cleaning for Large Databases," *United Nations Econ. Comm. Eur.*, 2008.

[44] R. K. Pearson, "The Problem of Disguised Missing Data," *ACM SIGKDD Explorations Newsletter*, vol. 8, no. 1, pp. 83–92, 2006.

[45] S. Krishnan, D. Haas, M. J. Franklin, E. Wu, and U. C. Berkeley, "Towards Reliable Interactive Data Cleaning : A User Survey and Recommendations," in *HILDA '16: Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 2016, no. 1, pp. 1–5, doi: https://doi.org/10.1145/2939502.2939511.

[46] F. Naumann, "Data Profiling Revisited," *ACM SIGMOD Rec.*, no. February, 2014, doi: https://doi.org/10.1145/2590989.2590995.

[47] Z. Abedjan, L. Golab, F. Naumann, and T. Papenbrock, *Data Profiling*. Morgan & Claypool Publishers, 2018.

[48] Z. Abedjan, L. Golab, and F. Naumann, "Profiling Relational Data – A Survey," *VLDB J. — Int. J. Very Large Data Bases*, no. June, 2015, doi: https://doi.org/10.1007/s00778-015-0389-y.

[49] Z. Abedjan, L. Golab, and F. Naumann, "Data Profiling – A Tutorial," *SIGMOD '17 Proc. 2017 ACM Int. Conf. Manag. Data*, no. October, 2017, doi: https://doi.org/10.1145/3035918.3054772.

[50] C. Yan and Y. He, "Synthesizing Type-Detection Logic for Rich Semantic Data Types using Open-source Code," *SIGMOD '18 Proc. 2018 Int. Conf. Manag. Data*, pp. 35–50, 2018, doi: https://doi.org/10.1145/3183713.3196888.

[51] M. Hulsebos, K. Hu, M. Bakker, E. Zgraggen, and T. Kraska, "Sherlock : A Deep Learning Approach to Semantic Data Type Detection," *KDD '19 Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1500–1508, 2019, doi: https://doi.org/10.1145/3292500.3330993.

[52] D. Zhang, M. Hulsebos, and W. Tan, "Sato : Contextual Semantic Type Detection in Tables," *KDD '19 Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019.

[53] N. Kanya and T. Ravi, "Modelings and Techniques in Named Entity Recognition- An Information Extraction Task," in *IET Chennai 3rd International on Sustainable Energy and Intelligent Systems (SEISCON 2012)*, 2012, pp. 1–5, doi: https://doi.org/10.1049/cp.2012.2199.

[54] V. Yadav and S. Bethard, "A Survey on Recent Advances in Named Entity Recognition from Deep," 2018.

[55] J. Li, A. Sun, J. Han, and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *Named entity Recognit. is task to identify mentions rigid Des. from text Belong. to predefined Semant. types such as Pers. Locat. Organ. etc. NER always serves as Found. many Nat. Lang. Appl. su*, p. 20, 2020.

[56] P. Krataithong, M. Buranarach, and N. Hongwarittorrn, "A Framework for Linking RDF Datasets for Thailand," in *Lecture Notes in Computer Science*, 2016, vol. 1, pp. 257–268, doi: 10.1007/978-3-319-49304-6.

[57] A. S. Zia, "A Survey on Different Searching Algorithms," *Int. Res. J. Eng. Technol.*, vol. 07, no. 01, 2020.

[58] H. Lim and N. Lee, "Survey and Proposal on Binary Search Algorithms for Longest Pre fi x Match," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 3, pp. 681–697, 2012.

[59] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis, "Conditional Functional Dependencies for Capturing Data Inconsistencies," *ACM Trans. Database Syst.*, vol. 33, no. 2, 2008, doi: 10.1145/1366102.1366103.

[60] X. Chu, I. F. Ilyas, and P. Papotti, "Holistic Data Cleaning : Putting Violations Into Context," in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 2013, doi: 10.1109/ICDE.2013.6544847.

[61] X. Chu, I. F. Ilyas, and P. Papotti, "Discovering Denial Constraints," *Proc. VLDB Endow.*,

vol. 6, no. 13, pp. 1498–1509, 2013, doi: https://doi.org/10.14778/2536258.2536262.

[62] T. Bleifuß, S. Kruse, and F. Naumann, "Efficient Denial Constraint Discovery with Hydra," *Proc. VLDB Endow.*, pp. 311–323, 2017, doi: https://doi.org/10.14778/3157794.3157800.

[63] Z. Wei, U. Leck, and S. Link, "Discovery and ranking of embedded uniqueness constraints," *Proc. VLDB Endow.*, vol. 2, no. 3, 2018, doi: https://doi.org/10.14778/3358701.3358703.

[64] F. Ferrarotti and S. Woltran, Eds., *Foundations of Information and Knowledge Systems*. Springer International Publishing, 2018.

[65] D. Hawking, *Identification of Outliers*. Springer Netherlands, 1980.

[66] V. Barnett, *Outliers in Statistical Data*. J. Wiley & Sons, 1994.

[67] V. J. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artif. Intell. Rev.*, 2004, doi: 10.1007/s10462-004-4304-y.

[68] J. Zhang, "Advancements of Outlier Detection: A Survey," *ICST Trans. Scalable Inf. Syst.*, 2013, doi: 10.4108/trans.sis.2013.01-03.e2.

[69] C. C. Aggarwal, *Outlier Analysis*. 2017.

[70] K. G. Mehrotra, C. K. Mohan, and H. Huang, *Anomaly Detection Principles and Algorithms*. 2017.

[71] C. D'Urso, "EXPERIENCE: Glitches in databases, how to ensure data quality by outlier detection techniques," *J. Data Inf. Qual.*, vol. 7, no. 3, 2016, doi: 10.1145/2950109.

[72] J. Hipp, U. Güntzer, and U. Grimmer, "Data Quality Mining - Making a Virtue of Necessity," *Proc. 6Th Acm Sigmod Work. Res. Issues Data Min. Knowl. Discov. (Dmkd 2001*, 2001, doi: 10.1111/j.1445-2197.2006.03638.x.

[73] S. Farzi and A. B. Dastjerdi, "Data Quality Measurement using Data Mining," *Int. J. Comput. Theory Eng.*, vol. 2, no. 1, pp. 115–118, 2009, doi: 10.7763/ijcte.2010.v2.125.

[74] F. Grüning, "Data quality mining: Employing classifiers for assuring consistent datasets," *Environ. Sci. Eng. (Subseries Environ. Sci.*, 2007, doi: 10.1007/978-3-540-71335-7_11.

[75] W. Dai, K. Yoshigoe, and W. Parsley, "Improving data quality through deep learning and statistical models," in *Advances in Intelligent Systems and Computing*, 2018, doi: 10.1007/978-3-319-54978-1_66.

[76] S. Gupta and A. Gupta, "Dealing with Noise Problem in Machine Learning Data-sets: A Systematic Review," *Procedia Comput. Sci.*, vol. 161, pp. 466–474, 2019, doi: 10.1016/j.procs.2019.11.146.

[77] J. D. Van Hulse and T. M. Khoshgoftaar, "The pairwise attribute noise," *Knowl. Inf. Syst.*, vol. 11, pp. 171–190, 2007, doi: 10.1007/s10115-006-0022-x.

[78] X. Zhu and X. Wu, "Class Noise vs . Attribute Noise : A Quantitative Study of Their Impacts," *Artif. Intell. Rev.*, no. 22, pp. 177–210, 2004, doi: https://doi.org/10.1007/s10462-004-0751-8.

[79] J. Kubica and A. Moore, "Probabilistic Noise Identification and Data Cleaning," *Third IEEE Int. Conf. Data Min.*, no. October, p. 19, 2003, doi: 10.1109/ICDM.2003.1250912.

[80] Y. Hu, S. De, Y. Chen, and K. Subbarao, "Bayesian Data Cleaning for Web Data," *ArXiv*, vol. abs/1204.3, p. 6, 2012.

[81] Z. Wei and S. Link, "DataProf : Semantic Profiling for Iterative Data Cleansing and Business Rule Acquisition," *SIGMOD '18 Proc. 2018 Int. Conf. Manag. Data*, pp. 1793–1796, 2018, doi: https://doi.org/10.1145/3183713.3193544.

[82] F. De Marchi and J. Petit, "Semantic sampling of existing databases through informative Armstrong databases," *Inf. Syst.*, vol. 32, no. 3, pp. 446–457, 2007, doi: https://doi.org/10.1016/j.is.2005.12.007.

[83] T. Papenbrock, T. Bergmann, M. Finke, J. Zwiener, and F. Naumann, "Data Profiling with

Metanome," in *Proceedings of the VLDB Endowment*, 2015, vol. 8, no. 12, pp. 1860–1863, doi: https://doi.org/10.14778/2824032.2824086.

[84] S. Krishnan and E. Wu, "AlphaClean: Automatic Generation of Data Cleaning Pipelines," no. July 2017, Apr. 2019.

[85] S. De, Y. Hu, Y. Chen, and S. Kambhampati, "BayesWipe: A multimodal system for data cleaning and consistent query answering on structured bigdata," *Proc. - 2014 IEEE Int. Conf. Big Data, IEEE Big Data 2014*, pp. 15–24, 2015, doi: 10.1109/BigData.2014.7004207.

[86] S. Das *et al.*, "Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2017, doi: 10.1145/3035918.3035960.

[87] M. Yakout, A. K. Elmagarmid, J. Neville, and M. Ouzzani, "GDR: A system for guided data repair," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2010, doi: 10.1145/1807167.1807325.

[88] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré, "HoloClean: Holistic Data Repairs with Probabilistic Inference," *Proc. VLDB Endow.*, vol. 10, no. 11, pp. 1190–1201, 2017, doi: 10.14778/3137628.3137631.

[89] M. Stonebraker *et al.*, "Data curation at scale: The data tamer system," in *CIDR 2013 - 6th Biennial Conference on Innovative Data Systems Research*, 2013.

[90] M. Mahdavi, F. Neutatz, L. Visengeriyeva, and Z. Abedjan, "Towards automated data cleaning workflows," *LWDA 2019 Proc. Conf. "Lernen, Wissen, Daten, Anal.*, vol. 2454, pp. 10–19, 2019.

[91] M. Mahdavi and Z. Abedjan, "REDS : Estimating the Performance of Error Detection Strategies Based on Dirtiness Profiles," *SSDBM '19 Proc. 31st Int. Conf. Sci. Stat. Database Manag.*, pp. 193–196, 2019, doi: https://doi.org/10.1145/3335783.3335808.

[92] S. Madden, Z. Abedjan, R. C. Fernandez, N. Tang, and M. Stonebraker, "Raha : A Configuration-Free Error Detection System," *SIGMOD '19 Proc. 2019 Int. Conf. Manag. Data*, pp. 865–882, 2019, doi: https://doi.org/10.1145/3299869.3324956.

[93] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer, "Profiler : Integrated Statistical Analysis and Visualization for Data Quality Assessment," in *AVI '12: Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012, pp. 547–554, doi: https://doi.org/10.1145/2254556.2254659.

[94] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, K. Goldberg, and U. C. Berkeley, "ActiveClean : Interactive Data Cleaning For Statistical Modeling," *Proc. VLDB Endow.*, vol. 9, no. 12, pp. 948–959, 2016, doi: https://doi.org/10.14778/2994509.2994514.

[95] A. A. Qahtan, A. Elmagarmid, R. C. Fernandez, and N. Tang, "FAHES : A Robust Disguised Missing Values Detector," in *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2100–2109, doi: https://doi.org/10.1145/3219819.3220109.

[96] R. Wirth, "CRISP-DM : Towards a Standard Process Model for Data Mining," in *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 2000, pp. 29–39, doi: 10.1.1.198.5133.

[97] S. Huber, H. Wiemer, D. Schneider, and S. Ihlenfeldt, "DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model," *Procedia CIRP*, vol. 79, pp. 403–408, 2019, doi: https://doi.org/10.1016/j.procir.2019.02.106.

5 Anexos

5.1 Anexo I – Conjuntos de datos empleados

A continuación se indican las referencias de los conjuntos de datos empleados en el trabajo, junto con capturas de pantalla de la herramienta de exploración desarrollada para visualizar los datos y los resultados obtenidos. Para una visualización interactiva y detallada de los datos se recomienda emplear dicha herramienta, la cual puede ejecutarse desde un explorador web abriendo el archivo `data-cleaning\wa\index.html`

Nombre	URL
Names_2010Census	https://www.census.gov/topics/population/genealogy/data/2010_surnames.html
athlete_events	https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results?select=athlete_events.csv
world-cities	https://datahub.io/core/world-cities
worldcities	https://www.kaggle.com/viswanathanc/world-cities-datasets?select=worldcities.csv

Tabla 5-1. *Referencias de los conjuntos de datos empleados.*

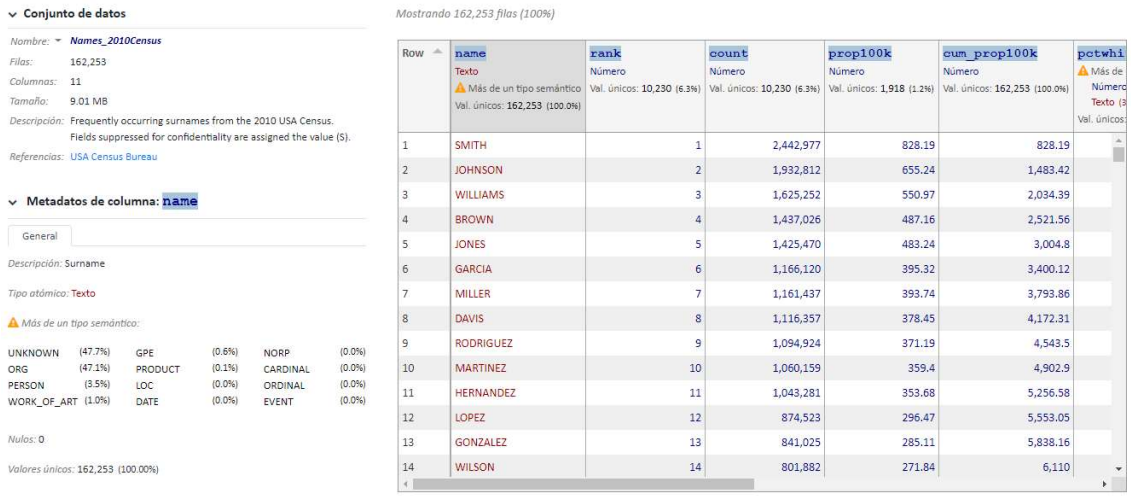
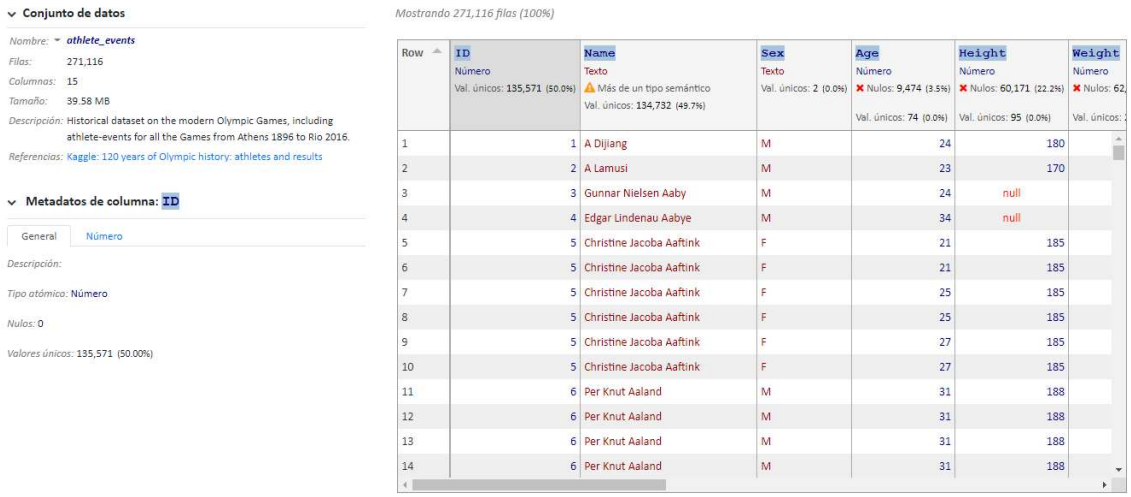


Figura 5-1. Captura de pantalla de la herramienta de exploración de datos y visualización de resultados del conjunto de datos *Names_2010Census*.



Conjunto de datos

Nombre: ***athlete_events***
Filas: 271,116
Columnas: 15
Tamaño: 39.58 MB
Descripción: Historical dataset on the modern Olympic Games, including athlete-events for all the Games from Athens 1896 to Rio 2016.
Referencias: Kaggle: 120 years of Olympic history: athletes and results

Metadatos de columna: ID

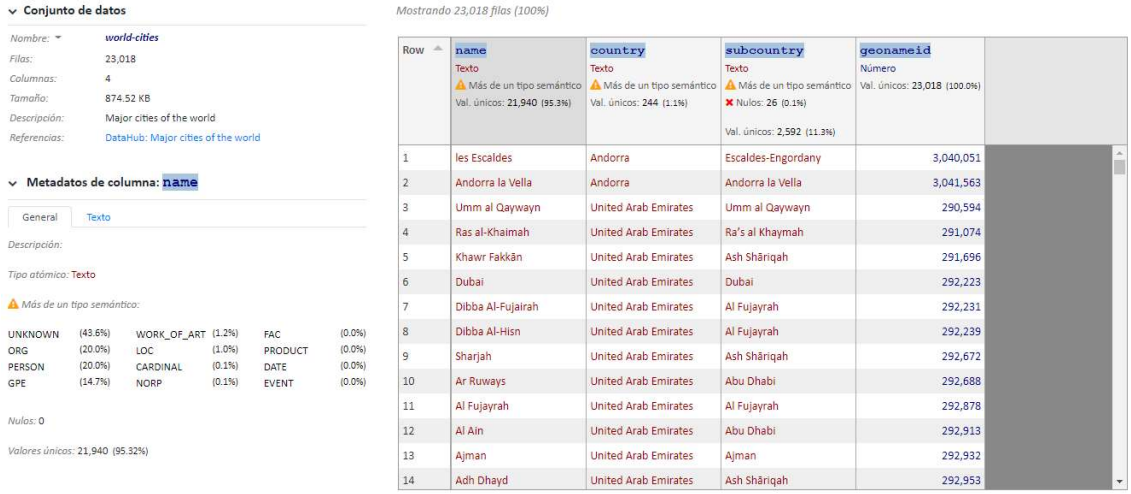
General | Número

Descripción:
Tipo atómico: Número
Nulos: 0
Valores únicos: 135,571 (50.00%)

Mostrando 271,116 filas (100%)

Row	ID Número Val. únicos: 135,571 (50.0%)	Name Texto ⚠ Más de un tipo semántico Val. únicos: 134,732 (49.7%)	Sex Texto Val. únicos: 2 (0.0%)	Age Número ✖ Nulos: 9,474 (3.5%) Val. únicos: 74 (0.0%)	Height Número ✖ Nulos: 60,171 (22.2%) Val. únicos: 95 (0.0%)	Weight Número ✖ Nulos: 62, Val. únicos:
1	1	A Dijiang	M	24	180	
2	2	A Lamusi	M	23	170	
3	3	Gunnar Nielsen Aaby	M	24	null	
4	4	Edgar Lindenau Aabye	M	34	null	
5	5	Christine Jacoba Aaftink	F	21	185	
6	5	Christine Jacoba Aaftink	F	21	185	
7	5	Christine Jacoba Aaftink	F	25	185	
8	5	Christine Jacoba Aaftink	F	25	185	
9	5	Christine Jacoba Aaftink	F	27	185	
10	5	Christine Jacoba Aaftink	F	27	185	
11	6	Per Knut Aaland	M	31	188	
12	6	Per Knut Aaland	M	31	188	
13	6	Per Knut Aaland	M	31	188	
14	6	Per Knut Aaland	M	31	188	

Figura 5-2. Captura de pantalla de la herramienta de exploración de datos y visualización de resultados del conjunto de datos *athlete_events*.



Conjunto de datos

Nombre: ***world-cities***
Filas: 23,018
Columnas: 4
Tamaño: 874.52 KB
Descripción: Major cities of the world
Referencias: DataHub: Major cities of the world

Metadatos de columna: name

General | Texto

Descripción:
Tipo atómico: Texto
⚠ *Más de un tipo semántico:*

UNKNOWN	(43.6%)	WORK_OF_ART	(1.2%)	FAC	(0.0%)
ORG	(20.0%)	LOC	(1.0%)	PRODUCT	(0.0%)
PERSON	(20.0%)	CARDINAL	(0.1%)	DATE	(0.0%)
GPE	(14.7%)	NORP	(0.1%)	EVENT	(0.0%)

Nulos: 0
Valores únicos: 21,940 (95.32%)

Mostrando 23,018 filas (100%)

Row	name Texto ⚠ Más de un tipo semántico Val. únicos: 21,940 (95.3%)	country Texto ⚠ Más de un tipo semántico Val. únicos: 244 (1.1%)	subcountry Texto ⚠ Más de un tipo semántico ✖ Nulos: 26 (0.1%) Val. únicos: 2,592 (11.3%)	geonameid Número Val. únicos: 23,018 (100.0%)
1	les Escaldes	Andorra	Escaldes-Engordany	3,040,051
2	Andorra la Vella	Andorra	Andorra la Vella	3,041,563
3	Umm al Qaywayn	United Arab Emirates	Umm al Qaywayn	290,594
4	Ras al-Khaimah	United Arab Emirates	Ra's al Khaymah	291,074
5	Khawr Fakkān	United Arab Emirates	Ash Shāriqah	291,696
6	Dubai	United Arab Emirates	Dubai	292,223
7	Dibba Al-Fujairah	United Arab Emirates	Al Fujayrah	292,231
8	Dibba Al-Hisn	United Arab Emirates	Al Fujayrah	292,239
9	Sharjah	United Arab Emirates	Ash Shāriqah	292,672
10	Ar Ruways	United Arab Emirates	Abu Dhabi	292,688
11	Al Fujayrah	United Arab Emirates	Al Fujayrah	292,878
12	Al Ain	United Arab Emirates	Abu Dhabi	292,913
13	Ajman	United Arab Emirates	Ajman	292,932
14	Adh Dhayd	United Arab Emirates	Ash Shāriqah	292,953

Figura 5-3. Captura de pantalla de la herramienta de exploración de datos y visualización de resultados del conjunto de datos *world-cities*

Figura 5-4. Captura de pantalla de la herramienta de exploración de datos y visualización de resultados del conjunto de datos *worldcities*

5.2 Anexo II – Herramienta para la exploración de datos y visualización de resultados

Como se ha mencionado varias veces a lo largo del presente trabajo, y en especial en la sección 1.8.7, se desarrolló una versión alfa de una aplicación web que permite la exploración de los conjuntos de datos y la visualización de los resultados de las técnicas empleadas. La aplicación está desarrollada en HTML/CSS/JavaScript empleando librerías de código abierto, y se accede ejecutando el archivo `data-cleaning\wa\index.html`. A grandes rasgos, el usuario selecciona un conjunto de datos, y luego la aplicación presenta de manera interactiva en tablas y gráficos los datos, metadatos y resultados de las técnicas empleadas, información que fue previamente generada por un script de Python. La siguiente figura esquematiza lo descrito.

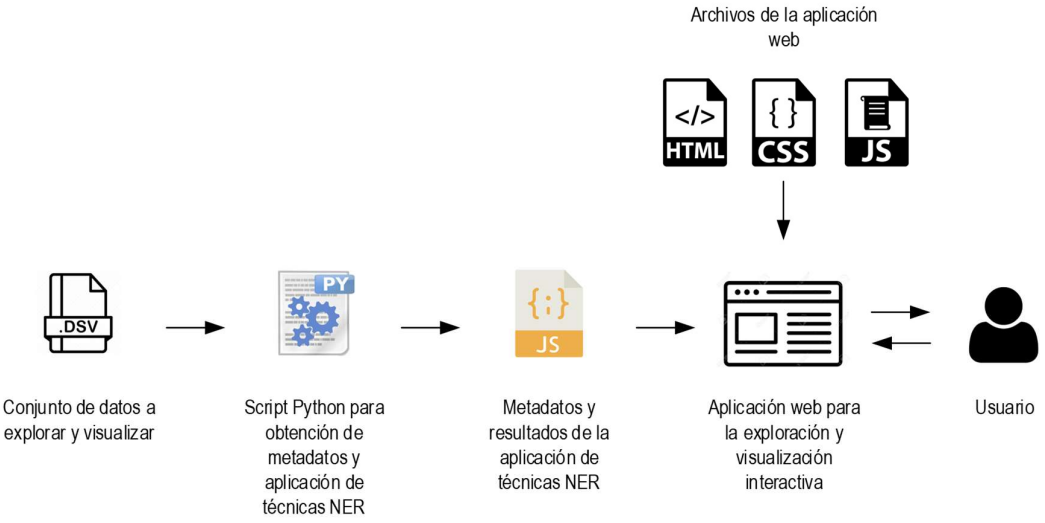


Figura 5-5. *Arquitectura de alto nivel de la aplicación web para la exploración de datos y visualización de resultados*.

La siguiente figura muestra la pantalla principal de la herramienta.

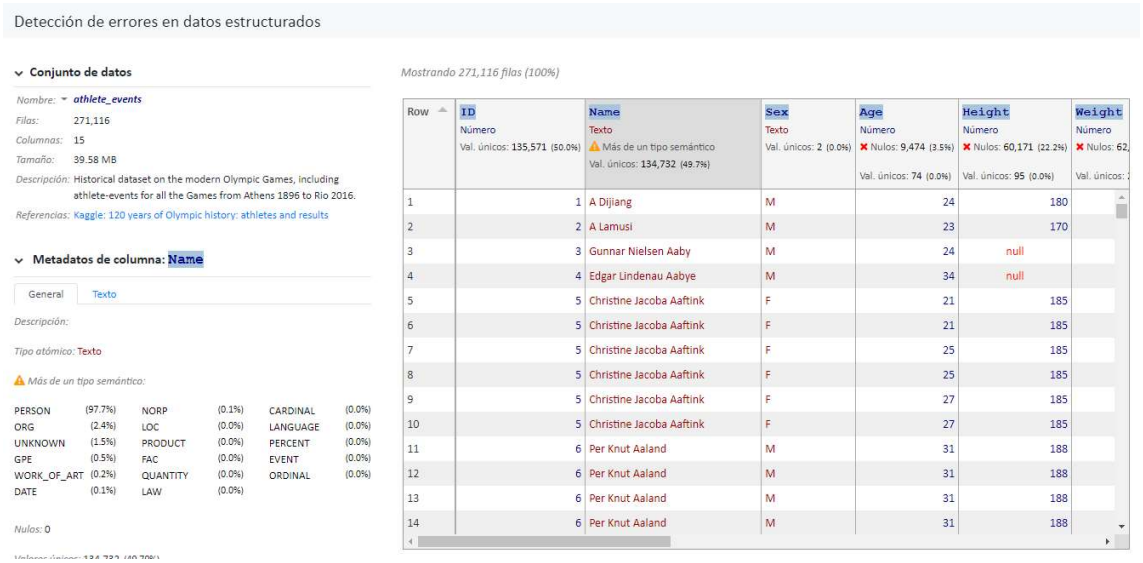


Figura 5-6. Pantalla de principal de la aplicación, donde puede observarse las secciones de información de resumen del conjunto de datos, una visualización en formato tabular de los mismos, y una sección con el detalle de los metadatos de la columna que se seleccione.

5.3 Anexo III – Duración promedio de la búsqueda binaria

La estimación del tiempo promedio necesario para buscar una cadena de texto empleando búsqueda binaria, se realizó tomando el promedio de la duración que demandó buscar mil veces en una lista de más de diez millones de elementos –10,207,305– los valores iguales al primer y último elemento de dicha lista. El detalle se puede encontrar en el script de Python `data-`

`cleaning\experiments\d00-imdb-names.py`, mientras que en la siguiente imagen se observa que el tiempo promedio obtenido fue de 1.7µs.

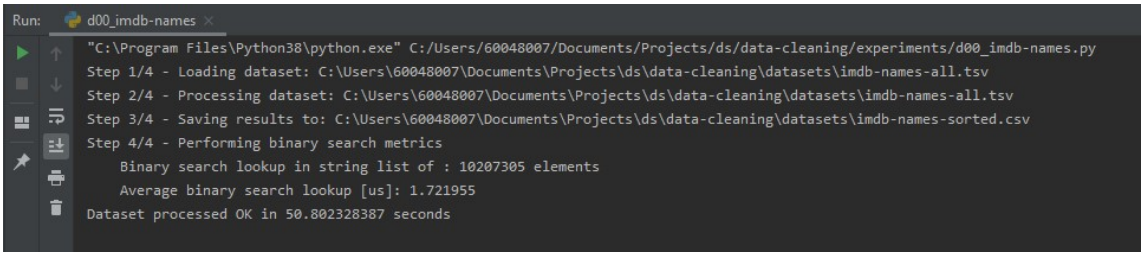


Figura 5-7. *Captura de pantalla con los resultados obtenidos al ejecutar el script d00-imdb-names.py*. Se observa que el tiempo promedio de buscar un valor de texto con el algoritmo de búsqueda binaria en una lista de más de 10 millones de elementos es de 1.7µs.

La lista donde se realizó la prueba fue construida con los nombres de los actores de las películas registradas por el sitio IMDB[40]: https://datasets.imdbws.com/name.basics.tsv.gz.

[40] IMDB: Internet movie database, https://www.imdb.com/