UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

Grupo de Investigación de Web Semántica

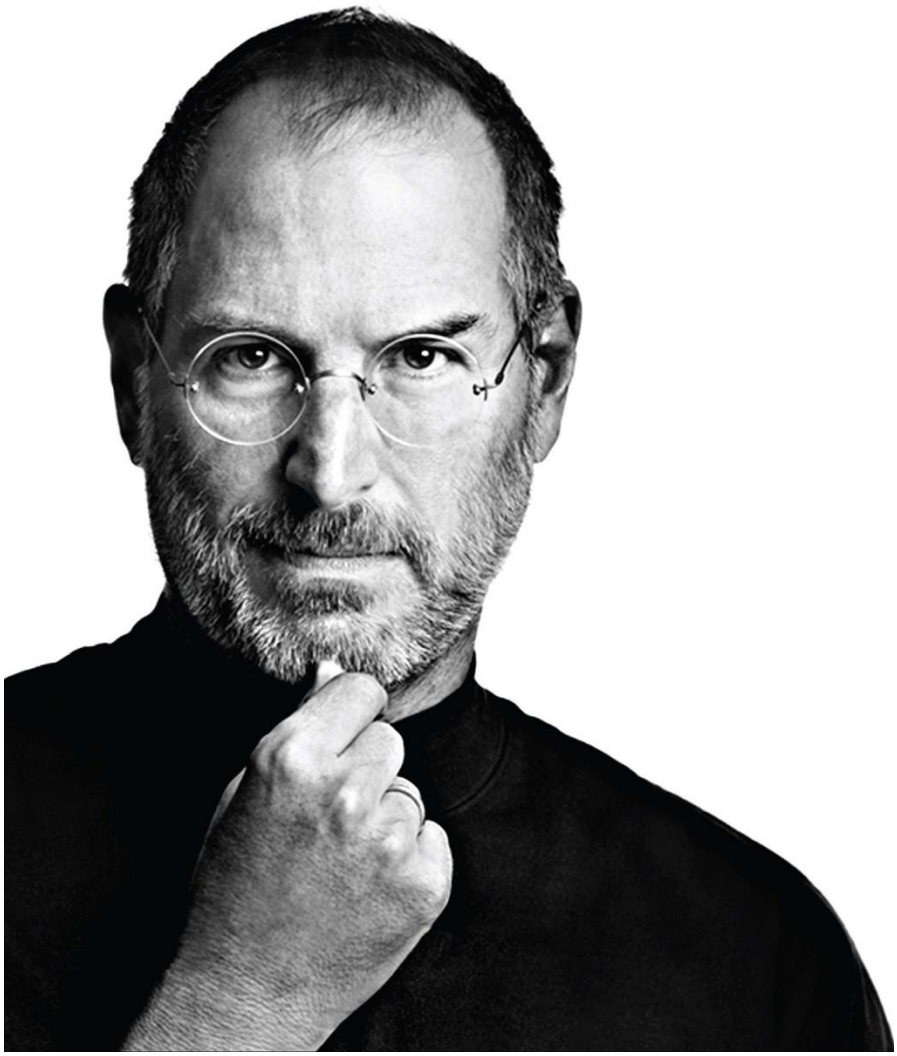
Método para la extracción, limpieza y publicación de datos de energía solar.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor:
Tahimí González Oliva

Tutor:
MSc. Mailen Edith Escobar Pompa
Ing. Alejandro Jesús Mariño Molerio

La Habana, junio de 2019
"Año 61 de la Revolución"



``Aquellos que están locos como para pensar que pueden cambiar el mundo, son aquellos que lo hacen´´.

Steve Jobs.

DECLARACIÓN DE AUTORÍA

Declaro ser la autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

_____	_____
Tahimí González Oliva	Alejandro Jesús Mariño Molerio
Autor	Tutor

_____

Mailen Edith Escobar Pompa

Tutor

DATOS DE CONTACTO

Síntesis del Tutor

Alejandro Jesús Mariño Molerio. Ingeniero en Ciencias Informáticas, de la Universidad de las Ciencias Informáticas, 2016. Profesor Instructor de la disciplina Programación e Inteligencia Artificial de la Universidad de las Ciencias Informáticas. Miembro del grupo de Investigación de Web Semántica desde el 2014. Líder de desarrollo en proyecto "Extracción, publicación y consumo de metadatos bibliográficos como datos enlazados". Miembro de la Asociación Cubana de Reconocimiento de Patrones (ACRP) desde el 2018. Autor y co-autor de varias publicaciones en revistas y memorias de eventos. Profesor de dos cursos de postgrado. Tutor de 3 Tesis de pre-grado para Ingeniería.

Mailen Edith Escobar Pompa. Ingeniera en Ciencias Informáticas, de la Universidad de las Ciencias Informáticas, 2008. Máster en Gestión de Proyectos Informáticos, 2016, Universidad de las Ciencias Informáticas, Cuba. Profesor Asistente de la disciplina Inteligencia Artificial de la Universidad de las Ciencias Informáticas. Miembro del grupo de Investigación de Gestión de Proyectos Informáticos desde el 2013. Planificadora de varios proyectos de los centros de desarrollos de Gestión Empresarial y Gobierno Electrónico 2008 – 2015. Miembro de la Asociación Cubana de Reconocimiento de Patrones (ACRP) desde el 2016. Autora y co-autora de varias publicaciones en revistas y memorias de eventos. Profesora de un curso de postgrado. Tutor de 5 Tesis de pre-grado para Ingeniería.

DEDICATORIA

A mis padres por su dedicación y apoyo incondicional. A mi hermano por la confianza, a mi sobrino por ser mi impulso a superarme.

AGRADECIMIENTOS

A mis profesores, en especial a la profesora Dariela.

A mis tutores por la dedicación y la ayuda brindada, en especial a mi tutor Alejandro Mariño por las horas dedicadas y apoyo.

Agradecer en especial a mis padres, a mi mamá por su apoyo incondicional, por la paciencia, por ser mi alma gemela. A mi padre por ser mi ejemplo a seguir, por ser mi guía. A mi hermano por la confianza y el apoyo.

A mi mejor amiga Migyara por su apoyo, compañía, dedicación y su sincera amistad.

A Marco por su compañía y apoyo en momentos difíciles.

A mis amigas de apartamento por su compañía y su amistad.

A mi amigo Pablo.

A mis compañeros.

RESUMEN

Las instituciones cubanas encargadas de las inversiones en energía solar se encuentran en desarrollo, como es el caso del Centro de Gestión de la Información y Desarrollo de Energía. Esta empresa para invertir en tecnologías de energía solar efectúa un estudio preliminar que se lleva a cabo de manera manual para calcular su factibilidad. Por otra parte, no presenta sus datos publicados en la web, lo que trae consigo que otras instituciones que de igual forma trabajan en estrategias para desarrollar en este tipo de energía, no dispongan de esos datos para sus inversiones. Todo esto trae consigo que se afecta el proceso de toma de decisiones con respecto a este tema. Con la presente investigación se desarrolla un método de limpieza que verifica la calidad de los datos. El método se estructura en tres fases fundamentales: (1) extracción, (2) limpieza y (3) publicación. Además, se genera un prototipo funcional que implementa el método propuesto. El desarrollo de esta investigación ayuda a la mejora del proceso de toma de decisiones para el uso de este tipo de energía en la empresa.

Palabras claves: energía solar, extracción de datos, limpieza de datos, publicación de datos.

ABSTRACT

Cuban institutions are responsible for investments in solar energy´s development, as is the case of Centro de Gestión de la Información y Desarrollo de Energía (CUBAENERGÍA). This company to invest in solar energy technologies make a preliminary study to calculate its feasibility. This study is make manually. Otherwise, it does not publish data on the web, which means that other institutions that likewise work on strategies to develop this type of energy do not have these data for their investments. All this entails affecting the decision-making process regarding this issue. The aim of the present research is to develop a cleaning method that verifies the quality of the data. The method is structured in three fundamental phases: (1) extraction, (2) cleaning and (3) publication. In addition, a functional prototype is implement. The development of this research helps to improve the decision-making process for this type of energy in the company.

Keywords: *data extraction, data cleaning, data publishing, solar energy.*

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En septiembre del 2015, las Naciones Unidas aprobaron la agenda 2030 sobre el desarrollo sostenible, firmada por todos los mandatarios en todo el mundo. La agenda 2030 consta de 17 objetivos de Desarrollo Sostenible, que brindan diferentes acciones en áreas de importancia crítica para la humanidad y el planeta. Los objetivos de Desarrollo Sostenible constituyen una herramienta de planificación para los países tanto a nivel nacional como local. El objetivo 7 plantea: “Garantizar el acceso a energía asequible, segura, sostenible y moderna para todos”. Las metas fundamentales para el año 2030 son el aumento de la proporción de energías renovables y la cooperación internacional, para facilitar el acceso de la investigación y las tecnologías relativas a la energía limpia, incluidas las energías renovables, la eficiencia energética y las tecnologías avanzadas y menos contaminantes de combustibles fósiles (Naciones Unidas 2015).

El término de energía renovable también definida como energía limpia o alternativa, en teoría no se agota con el paso del tiempo, debido a que es resultado de la energía que llega del planeta de manera continua. Las mismas no generan residuos de difícil eliminación. Su impacto ambiental es reducido, pues no producen emisiones de dióxido de carbono y/o gases contaminantes a la atmósfera, como es el caso de los combustibles fósiles, que crean emisiones de gases de efecto invernadero que contribuyen al calentamiento global y al cambio climático (Alvarado 2013).

Las energías renovables son capaces de regenerarse fácilmente, de forma natural o artificial, pues provienen de fuentes como el sol, el agua, el viento, el mar, recursos biológicos, el suelo, entre otras fuentes. Por otra parte, son una elección viable para la economía y la sociedad, pues resultan rentables en la generación de la electricidad, a pesar de que reducen no solo los costos ambientales sino también los económicos en su implementación y uso, aun así, existen países que no tienen acceso a ellas económicamente. Estas diversifican la matriz de generación eléctrica, impulsan la competitividad en el mercado y son de fácil adquisición por los consumidores y/o usuario. Existen varios tipos de energías renovables, ejemplo de ellas son, la energía geotérmica, la solar, que contiene la fotovoltaica y la térmica, además existe la energía eólica, la hidráulica, la biomasa que a su vez tiene la biomasa residual, biocarburantes, cultivos energéticos, residuos sólidos urbanos (Isabel et al. 2017).

Muchos países de todo el mundo trabajan en estrategias nacionales para implementar la agenda 2030, creando y financiando infraestructura y tecnología relacionadas con fuentes de energía renovables. En el continente asiático, China ha logrado importantes mejoras en este sentido, este país avanza a pasos agigantados en las tecnologías y desarrolla energía solar y eólica, toma el liderazgo en la actualidad, se encuentra invirtiendo para el desarrollo tecnológico e investigativo relacionado con las fuentes de energía renovables, más de la mitad de la inversión está enfocada en la energía solar. Por otro lado,

se encuentra el continente europeo, antes líder del desarrollo de energía renovable, actualmente a pesar de ello sigue invirtiendo para su mejora.

Desde el triunfo de la Revolución, en Cuba se ha evidenciado un interés político en el fomento del uso de energías renovables. En la Política Económica y Social del Partido y la Revolución, aprobada en junio de 2014 por el Consejo de Ministro, se encuentran dos lineamientos que priorizan el desarrollo perspectivo de las energías renovables y las eficiencias energéticas. El 203 que plantea: "*la fomentación de la cogeneración y trigeneración en todas las actividades con posibilidades"*, y el 204 refiere: "*el aceleramiento del cumplimiento del programa aprobado hasta el 2030, para el desarrollo de dichas fuentes y el uso eficiencia de la energía".* Los objetivos fundamentales para el 2030 son:

1. El aumento del porcentaje de utilización de energías renovables a un 24%.
2. Reducir los costos de la energía entregada y la contaminación medioambiental.
3. Incrementar la dependencia de importaciones de combustibles para la generación.

Cuba es un país rico en recursos energéticos renovables, como son el sol, el viento, la biomasa (fundamentalmente la procedente de la caña de azúcar) y la hidroenergía. Producto a las condiciones geográfica que tiene el archipiélago, el gobierno cubano ha potenciado el desarrollo de las energías eólicas, fotovoltaica y de biomasas. Para lograr el cumplimiento de los objetivos de la Política Económica y Social del Partido y la Revolución, se necesita un período de transición donde se vayan introduciendo paulatinamente estas tecnologías en conjunto con un programa de medidas (ahorro, eficiencia energética y cogeneración). En el contexto del programa de medidas se han construido centrales hidroeléctricos, instalaciones de paneles fotovoltaicos, sistemas termo solar, y la utilización de otras fuentes como la eólica y biomasa (Cubaenergía 2015).

En el ámbito cubano existen varias instituciones dedicada al desarrollo de energías renovables. El Centro de Gestión de la Información y Desarrollo de la Energía (CUBAENERGÍA), perteneciente a la Agencia de Energía Nuclear y Tecnologías de Avanzada (AENTA) y adscrito al Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA), es una entidad pública presupuestada de investigación – desarrollo y servicios científico – técnicos en materia de energía y medio ambiente. Se encargada del fomento e inversión en energías renovables, siendo la energía solar una de las principales áreas de desarrollo en este sector.

Para el uso de la energía solar se realizan inversiones en sistema fotovoltaicos o calentadores solares para ello, se efectúa un estudio preliminar. Para determinar la factibilidad de dicha inversión se lleva a cabo un proceso eminentemente manual, esto implica demora en el resultado del mismo, además de que tiende a aumentar el grado de error o una deformidad de la información. Por otro parte, se hace

uso de los datos, tomando como referencia solo los propios datos de la empresa, sin tener en cuenta datos externos que generan otras entidades.

La empresa dispone de *dataset* en formato de hoja de cálculo que posee una estructura definida a priori por especialistas. En la hoja de cálculo además de los *raw data* o datos brutos existen datos derivados[1], que son utilizados para realizar cálculos en el proceso de toma de decisiones sobre radiación solar. Los datos contenidos en el *dataset* son resultado de mediciones realizadas, utilizando instrumentos especializados, para calcular la radiación solar en determinados puntos geográficos del archipiélago cubano. Las mediciones son adicionadas al *dataset* mediante un proceso manual efectuado por un especialista de la entidad.

Como resultado del proceso descrito anteriormente se tiene que la utilización de estos datos puede estar sujetos a la existencia de anomalías, datos sucios o incompletos y datos nulos, entre otros, dado que no existe un proceso donde se verifique la calidad de los datos almacenados una vez adicionados. La ausencia de este proceso de verificación y la forma en que se dirige el mismo, afecta el proceso de toma de decisiones.

Por otra parte, estos datos no se encuentran disponibles en un repositorio en la web para su uso por otras empresas afines a este sector. Para su utilización se debe solicitar una cita a la entidad, para luego entrevistarse con un especialista de la empresa encargado de este tema. Dicho especialista es el que provee los datos sobre energía solar.

En consecuencia, este engorroso proceso afecta de manera negativa la eficiencia en la toma de decisiones no sólo de la entidad sino de las entidades externas a las que le pueda ser útil dicha información.

De acuerdo con la situación descrita anteriormente, se plantea el siguiente **problema a resolver:** ¿Cómo mejorar la centralización y calidad de los datos de fuentes de energías renovables que permita apoyar la toma de decisiones en esta área en Cuba?

El objeto de estudio de la investigación se enmarca en el proceso de extracción, limpieza y publicación de datos.

En la presente investigación se propone como **objetivo general:** desarrollar un método para la extracción, limpieza y publicación de datos de fuentes de energía solar basados en tecnologías semánticas.

A partir de lo planteado anteriormente se desglosan los siguientes **objetivos específicos:**

[1] datos que se derivan de otros datos. Ej. Datos derivados como resultados de cálculos.

- Elaborar el marco teórico de la investigación sobre los procesos de extracción, limpieza y publicación de datos.
- Implementar un método que permita realizar la extracción, limpieza y publicación de datos de fuentes de energía solar.
- Validar el método propuesto mediante un caso de estudio.

Se obtienen como artefactos a entregar: método para la extracción, limpieza y publicación de fuentes de energía solar y prototipo funcional que implemente el método propuesto.

Campo de acción: extracción, limpieza y publicación de fuentes de datos de energía solar.

Se definen como tareas a cumplir para el desarrollo de la investigación:

- Revisión bibliográfica acerca de los procesos de extracción, limpieza y publicación de datos para determinar los principales conceptos y técnicas asociados a los mismos.
- Elaboración del marco teórico del tema de investigación.
- Diseñar un método para extracción, limpieza y publicación de datos de energía solar.
- Estudio de las herramientas para el desarrollo de algoritmos para la extracción, limpieza y publicación de datos.
- Análisis para definir las herramientas, métodos y metodologías de desarrollo de software para la implementación de los algoritmos.
- Implementación de los algoritmos para la extracción, limpieza y publicación de datos de fuentes de energía renovables.
- Validación de la propuesta de solución mediante la aplicación en un caso de estudio.

Se plantea como **idea a defender:**

Con el desarrollo de un método para la extracción, limpieza y publicación de los datos se mejorará la centralización y calidad de los datos de fuentes de energías solar que permita apoyar la toma de decisiones de esta área en Cuba.

A continuación, se definen los **métodos de investigación**:

Métodos Teóricos:

Método Analítico-Sintético: posibilita la realización del estudio de los principales conceptos de la investigación, ayudando en el proceso de análisis de los documentos y la extracción de los elementos más trascendentales a tener en cuenta para el desarrollo del método de limpieza e integración de datos.

Método de la modelación: permite modelar los aspectos con el objetivo de explicar la realidad del problema de investigación dando la medida en que se logra el objetivo de la misma.

Método de revisión sistemática de la bibliografía: posibilita una evaluación rigurosa y fiable de la investigación, presenta un resumen de evidencias a través de métodos de búsquedas sistemáticas comprensibles y síntesis de información seleccionada.

Métodos Empíricos:

Método de medición: Posibilita medir la calidad de los datos de energías renovables para obtener una medida de la calidad de los mismos en los distintos momentos de la investigación.

Método de entrevista: Posibilitó la recogida de información mediante la realización de una entrevista a un especialista de la empresa Cubaenergía, lo que permitió un aumento de la información asociada a la investigación y una medición de aspectos a tener en cuenta.

Estructura de la tesis

La tesis se estructura en tres capítulos.

Capítulo 1. **Fundamentación Teórica.** En este capítulo se abordan los conceptos asociados a la investigación para el desarrollo de la propuesta de solución, centrándose en el estudio de las temáticas de extracción, limpieza y publicación de datos.

Capítulo 2. **Propuesta de solución.** En este capítulo se describe el método para la realización de extracción, limpieza y publicación de datos de fuentes de energía solar para apoyar la toma de decisiones sobre esta área en Cuba. Además, se describen las herramientas a utilizar para el desarrollo de la solución.

Capítulo 3. Validación de la solución. En este capítulo se realiza la validación de la solución mediante la aplicación del método en un caso de estudio.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se utiliza el método de revisión sistemática de la bibliografía, el cual permite llevar a cabo la evaluación rigurosa y fiable dentro del tema de extracción, limpieza y publicación de datos. Por otro lado, se presenta un resumen de evidencias a través de métodos de búsquedas sistemáticas comprensibles y una síntesis de la información seleccionada.

El desarrollo del método se divide en tres etapas: planificación, ejecución y análisis de los resultados obtenidos, con el propósito de presentar un registro que ofrece la información correspondiente descrita para que otros investigadores dispongan de dicha investigación.

1.2. Planificación de la investigación

La etapa de planificación trata acerca de la definición de las preguntas científicas por las cuales se guía la investigación, como definición de palabras claves, criterios para la inclusión y exclusión, estrategias de búsqueda y artículos de calidad acerca de la investigación. Por lo tanto, los siguientes puntos definen las preguntas que estarán enfocadas a la investigación:

PI1: ¿Cuál es el resumen de las investigaciones actuales en limpieza de datos?

PI2: ¿Cuáles son las técnicas más utilizadas en limpieza de datos?

PI3: ¿Cuáles son los algoritmos más utilizados para la limpieza de datos?

PI4: ¿Cuáles son los trabajos de investigación más citados relacionados con la limpieza de datos?

PI5: ¿Cuáles son los principales enfoques para la extracción y publicación de datos?

PI6: ¿Cuál fue el resultado de las investigaciones anteriores?

PI7: ¿Qué impacto tendrá para Cuba el uso de algoritmos y métodos en la limpieza de datos para la toma de decisiones en el contexto de las fuentes de energías renovables?

A continuación, se presenta en la siguiente tabla los criterios de inclusión y exclusión utilizados para el estudio de la investigación, en el proceso de selección bibliográfica, con el objetivo de lograr una mejor calidad en el mismo.

Tabla I Criterios de inclusión, exclusión y calidad.

Inclusión	Estudios en inglés.
	Estudios significativos sobre la extracción, limpieza y publicación de datos.
	Artículos científicos, tesis de grados.
Exclusión	Estudios en español.
	Estudios fuera del contexto de la investigación.
	Estudios publicados antes del 2014.
Calidad	Estudios con resultados completos.

	Estudios con diferentes propuestas.

En esta fase para la revisión de la calidad se tiene en cuenta de cada artículo encontrado, el título, resumen, palabras claves, algoritmos propuestos, técnicas y resultados.

Para esto se decide el uso de las siguientes fuentes bibliográficas, para el procesamiento de la búsqueda de los documentos:

- IEEE Xplore Digital Library[2]
- ACM Digital Library[3]
- Science Direct[4]
- Springer[5]

Luego de haber seleccionado los métodos de búsqueda en las fuentes bibliográficas anteriormente citadas, se genera la siguiente cadena ((“dataset” AND solar energy”) AND (“data quality” OR “technique”))

((“data quality” OR “data extraction” OR “data publication” OR “data cleaning”)) AND (“methods” OR “method” OR “algorithm” OR “algorithms” OR “techniques” OR “technique”)), para identificar los documentos válidos a consultar.

1.3. Ejecución del Plan

En esta etapa de ejecución del plan se involucran cinco pasos: (1) realizar la búsqueda en las bases de datos seleccionadas; (2) comparación de resultados de búsquedas con la exclusión de artículos repetidos; (3) aplicación de la inclusión, exclusión, y criterios de calidad; (4) evaluación de todos los estudios que aprobó la revisión inicial; y (5) síntesis de datos.

La Figura 1 muestra el flujo la ejecución de las cadenas de búsquedas en las bases de datos, encontrándose un conjunto de 1131 de documentos. Para la revisión y el logro de una mejor precisión y fiabilidad se utilizó la herramienta StArt (Estado del Arte a través de Revisiones Sistemáticas), dicha herramienta tiene el propósito de apoyar la investigación en el análisis y una mayor calidad de los resultados.

[2] ieeexplore.ieee.org
[3] dl.acm.org
[4] www.sciencedirect.com
[5] www.springer.com

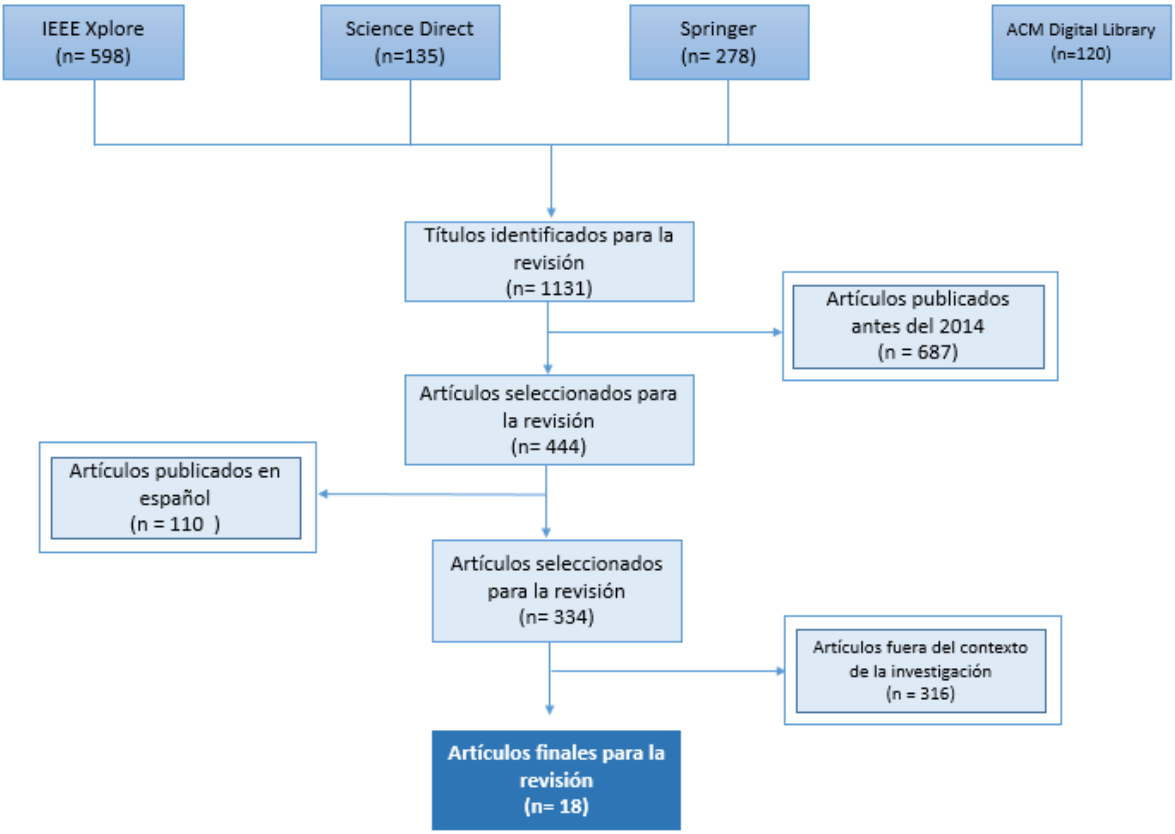


Figura 1 Proceso de selección de artículos.

Finalmente, después de la preselección de los trabajos, se realizó un análisis de los datos, con el objetivo de aplicar una evaluación basada en los criterios de calidad establecidos. De este modo, de los artículos, fueron eliminados 1113, lo que dio lugar a un conjunto final de 18 artículos de información relevantes para la investigación.

1.4. Resultados

En este subepígrafe se muestra los resultados del análisis de los artículos seleccionados. Por lo que a continuación aparece cada uno de los pasos realizados que responderán las preguntas planteadas al comienzo de este capítulo, las mismas son estructuradas de la siguiente manera: RPI (Respuesta para la Pregunta de la Investigación), el número y la pregunta.

RPI1: ¿Cuál es el resumen de las investigaciones actuales en limpieza de datos?

Los sistemas de información, en la actualidad están presentes en casi todas las esferas de la sociedad, y ha conllevado el procesamiento de enormes volúmenes de datos, que se interrelacionan continuamente entre cada una de estas esferas. En el procesamiento realizado por estos sistemas, en ocasiones se provocan anomalías en los resultados que se obtienen de estos procesos. Como posible solución a este problema y garantizando la eliminación de errores e inconsistencia, se realiza una búsqueda en los dataset (conjunto de datos) para su consecuente corrección. Para la detección y

eliminación de estos posibles errores, existen métodos y algoritmos computacionales encargados de estos procesos, que se agrupan bajo el término denominado **Limpieza de datos.**

- Limpieza de datos se define como el conjunto de métodos cuyo objetivo es detectar y eliminar errores e inconsistencia de los datos para mejorar su calidad, suavizando los datos ruidosos, identificando o eliminando los valores anómalos y resolviendo inconsistencias (Calabrese 2018).
- Otra definición dada por (DHRUV Gairola 2015) es que los sistemas de limpieza de datos funcionan automatizando la corrección de errores de datos. Estos se pueden basar en restricciones definidas en conjuntos de datos para detectar errores, corrigiéndolos así para un conjunto de datos resultante que satisfaga las restricciones ya definidas con anterioridad.

Como resultado del proceso de limpieza de datos, se pueden obtener datos con calidad por lo que mejora la capacidad de información consistente y confiable para las instituciones que hacen uso de los mismos. Por otro lado, a pesar de existir un proceso que limpia los datos y que garantiza la calidad de los mismos, se crea la necesidad de unificar varios dataset que difieren en cuanto a forma. Actualmente las organizaciones acumulan una gran cantidad de información empresarial, esta debe de ser llevada de una manera adecuada y eficiente para conocer el estado de la organización, sobre todo en la ola tecnológica en la que la mayoría se encuentra incursionando. Contar con información exacta y precisa es un recurso muy valioso para las empresas, ya que de ella dependen variables de crecimiento. A partir de una información se toma decisiones y se proyectan metas u objetivos. Uno de los objetivos básicos del manejo de información es apoyar la toma de decisiones de la gerencia, que ayudará al crecimiento, fortalecimiento y expansión de las mismas.

RPI2: ¿Cuáles son las técnicas más utilizadas en limpieza de datos?

Actualmente existen técnicas que se utilizan para solucionar los problemas de calidad de datos: inexactitud, inconsistencia e inaccesibilidad, todo esto se lleva a cabo debido a que los datos se encuentran ubicados en diferentes sistemas, la existencia de registros duplicaos, datos faltantes, datos incorrectos, datos obsoletos, entre otros problemas.

Entre las técnicas más utilizadas para la limpieza se encuentra **las reglas de negocio**: Restricciones que definen condiciones que deben ser aplicadas en situaciones específicas (dependen mucho de la organización y de los tipos de datos que esta posee)(Calabrese 2018).

Por otro lado la **técnica de detección de errores cualitativos**; se tienen en cuenta tres interrogantes (qué, cómo y dónde detectar el error). Para la misma se tiene una base de datos sucia para primeramente detectar los errores anómalos, teniendo en cuenta los tipos de errores, ya sea por inconsistencia en la integridad de los datos o por la duplicidad de los mismos, para la segunda interrogante se detecta ya sea manual o de forma automatizada y por último donde se detecta ya sea desde su origen o desde el objetivo a donde se quiere llegar.

Técnica de detección de anomalías: en esta se proponen tres enfoques para resolver el problema de anomalías, el primero es determinar los valores anómalos sin conocimiento previo de los datos, este enfoque procesa los datos como un archivo estático. El segundo enfoque es el que modela la normalidad y la anormalidad de los datos, es un enfoque análogo a la clasificación supervisada y requiere etiquetado previo de los datos (etiquetado normal y anormal), y por último el tercer enfoque se modela solo la normalidad y necesita datos preclasificados debido a que solo asume los datos marcados como normales, trabaja con datos estáticos y dinámicos (Cimiano y Bielefeld 2016).

Técnica de minería de patrones: Esta técnica está diseñada de tal manera que la supervisión humana no es obligatoria, la misma se basa en detectar errores que se producen de manera individual. La técnica de minería de patrones se utilizan típicamente para descubrir asociaciones positivas entre los ítems (Bertens 2015).

Técnica de reparación: Esta técnica se utiliza cuando lo datos son inconsistentes, por lo que usa restricciones de reparación para reducir esta inconsistencia y mejorar la calidad de los datos. Tiene como objetivo identificar las restricciones antiguas y encontrar nuevas restricciones para una mejora en el desarrollo de calidad de los datos. Diseñada para áreas donde los datos evolucionan dinámicamente, por lo que las restricciones están cambiando constantemente. Esta técnica suele utilizar un clasificador para recomendar el tipo de reparación necesaria para resolver la inconsistencia de los datos. (Volkovs y Chiang 2016).

De las técnicas descritas anteriormente no se selecciona para la investigación la reparación de errores, detección de errores y la técnica de minería de patrones. En el caso de la primera técnica solo se trabaja con datos inconsistentes y se enfoca en datos que evolucionan en el tiempo de manera dinámica, cualidad que no posee los datos que se manejan en durante la investigación. Para la segunda técnica se tiene en cuenta los datos duplicados, y los datos que se manejan en la investigación no es objetivo eliminarlos en caso que se encuentren duplicados. La tercera técnica se realiza solo para encontrar nuevas restricciones para mejorar la calidad de los datos, por lo que no detecta datos incorrectos o inconsistentes. Para el caso de la técnica de minería de patrones, se enfoca en detectar errores a través de patrones definidos, por lo que se intenta encontrar asociaciones entre los datos o patrones, por tanto tampoco detecta los datos que se desea. Por todo ello la técnica que se decide seleccionar es la técnica de detección de anomalías, debido a que el objetivo es encontrar la presencia de datos incorrectos y anómalos. Por otro lado se utiliza las reglas de negocio para precisar las restricciones que deben de cumplir los datos para tomarlos como datos normales, con respecto al negocio.

RP3: ¿Cuáles son los algoritmos más utilizados para la limpieza de datos?

Para la limpieza de datos se mencionan diferentes algoritmos. A continuación, se hace referencia a cada uno de ellos:

- **DYSNI: utiliza un árbol trenzado (BRT)**, para ordenar un conjunto de datos y trabajar con registros que se comparan dentro de un sistema de ventana. Trabaja además con datos estáticos debido a que no tiene presente la necesidad de ubicar el tamaño de ventana. Este método presenta la desventajada de no poder trabajar con datos en tiempo real o en base de datos dinámicas ya que tiene que ser trabajado con arreglos clasificados como rígidos (Ramadan, Christen y Liang 2014).
- **DCS++ (Estrategia de Conteo por Duplicados +++):** es una versión mejorada de SNM (Método de Vecindario Clasificado), que clasifica los datos de acuerdo a alguna clave y luego avanza una ventana sobre los datos comparando sólo los registros que aparecen en la misma ventana. El DCS++ utiliza un tamaño de ventana variable, obteniéndose los mismos resultados con menos comparaciones(Zhang, Chou y Churchill 2018).
- **InnWin (Ventanas Innovadoras):** este es una variante de SNM que aumenta un tamaño de ventana en cada duplicado detectado como DCS++ y termina cuando los sucesivos no duplicados superen un cierto límite. Este algoritmo asume que el registro de duplicados en orden, aumenta la probabilidad de encontrar más duplicados (Bano y Azam 2015).
- **Dedup (Deduplicación):** este se utiliza para mejorar la eficiencia del almacenamiento, siendo esto la clave para optimizar la capacidad. Dedup es utilizado para evaluar, identificar, rastrear y evitar la duplicidad incluyendo la actualización de la información de seguimiento (Wandhekar 2015).
- **K means:** según (Allahyari, Trippe y Gutierrez 2017) es uno de los algoritmos utilizado en la minería de datos. Las k-medias agrupan particiones. Una de sus principales características que lo hacen eficiente es que la distancia euclidiana se usa como una métrica y la varianza es usada como una medida de la dispersión de los grupos.
- **Z-score:** es un algoritmo de normalización, también llamado puntaje estándar. Se aplica comúnmente en los campos de estadística y trabajo con datos. El método o algoritmo Z-score se utiliza para evaluar cuantitativamente fallas, y detectar datos o estadísticas anómalas, estableciendo un coeficiente anormal, se desarrolla a partir de la desviación estándar (Wang et al. 2017).

A partir de las descripciones anteriores de los algoritmos se concluye que no se hará uso de los siguientes: DYSNI, DCS++, InnWin y Dedup, debido a que de manera general, estos algoritmos tienen como principal objetivo eliminar o detectar los datos duplicados ordenándolos en conjunto de datos. Por lo que se decide utilizar para el desarrollo de la investigación los algoritmos Z-Score y K-means ya

que estos se enfocan en detectar datos erróneos o incorrectos. En el caso del Z-Score se ajusta en el trabajo de datos, detectando datos anómalos, estableciendo un coeficiente de anomalías para la comparación de los datos. Por otro lado K-means se utiliza debido a que este calcula la diferencia de los datos a través de un centroide establecido, por lo que los datos que están fuera de área normal son detectados.

RPI4: ¿Cuáles son los trabajos de investigación más citados relacionados con la limpieza de datos?

A continuación, se muestran los artículos seleccionados:

Tabla II Cantidad de citas por artículos.

Artículo	Citas
[1] Data Cleaning	34
[2] Trends in Cleaning Relational Data	65
[3] Cleaning data with forbidden Itemsets	10
[4] Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods	221
[5] Continuous Data Cleaning	68
[6] Interactive and Deterministic Data Cleaning	27
[7] Detecting Anomalous	5
[8] A survey of network anomaly detection techniques	219

Con el estudio de estos documentos acerca del proceso de limpieza se obtuvo una amplia información sobre el tema, donde se puede concluir que dentro de los trabajos de investigación sobre limpieza de datos más citados se encuentra el artículo [4] Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods, [2] Trends in Cleaning Relational Data, el [5] Continuous Data Cleaning y el [8] A survey of network anomaly detection techniques.

PI5: ¿Cuáles son los principales enfoques para la extracción y publicación de datos?

Para la extracción de datos en una hoja de cálculo o Excel lo más recomendable es crear una base de datos. Para ello existen varios programas como es el caso del proceso de extracción, transformación y carga (*ETL*, por sus siglas en inglés), proceso para combinar datos de múltiples fuentes y convertirlos en un formato menos complejo para trabajar y cargarlos recomendablemente en una base de datos (Andersen, Thomsen y Torp 2018). Cuando se trata de grandes volúmenes de datos se recomienda

emigrar toda la información a una base de datos. Mediante el proceso de extracción se leen los datos para luego realizar la transformación, una vez extraídos y llevados a una base de datos, en el proceso de transformación se convierten para luego ser colocados a otra base de datos. Mientras que en el proceso de carga se escriben los datos en la base de datos de destino.

Por otra parte la publicación de datos se enfoca en la actualidad en la publicación en la web, por lo que lo más recomendable son los servicios web. Uno de los servicios web más utilizado es la Transferencia de Estado Representacional (*REST*, por sus siglas en inglés), servicio que remplazó a *SOAP* debido a los diversos grados de estandarización. *REST* ofrece una forma efectiva de interactuar con clientes ligeros. Además, ofrece las siguientes ventajas en comparación con *SOAP*:

- Los servicios SOAP son mucho más difíciles de escalar que los servicios *RESTfull*, por lo tanto *REST* se elige a menudo como la arquitectura de servicios.
- *REST* utiliza un formato de mensaje más pequeño que *SOAP*. *SOAP* utiliza XML para todos los mensajes, lo que hace que el tamaño del mensaje sea mucho mayor y por lo tanto menos eficiente, esto quiere decir que *REST* proporciona un mejor rendimiento, así como la reducción de los costos con el tiempo.
- *REST* puede hacer uso de las aplicaciones más rápido de lo que lo puede hacer *SOAP*.

Por otra parte, en la actualidad ha crecido el uso de *GraphQL* en comparación con REST. Este aumento en su uso es debido a que provee de una selección de los datos en las respuestas a las peticiones, es fuertemente tipado lo que permite la detección de errores durante su desarrollo y posee un buen rendimiento que se ve reflejado en la escalabilidad como una característica intrínseca de esta arquitectura de servicios.

Sin embargo, en la investigación se escoge REST, que, aunque no sea fuertemente tipado y en la respuesta devuelva todos los datos, según los intereses y objetivos de la investigación, esta arquitectura de servicio se ajusta a las necesidades requeridas. Estas se centran en proporcionar un servicio web confiable, sencillo y que sea escalable teniendo en cuenta la infraestructura tecnológica actualmente disponible por el cliente.

PI6: ¿Cuál fue el resultado de las investigaciones anteriores?

Luego de analizar los trabajos resultantes de la revisión sistemática se da como resultado que para el trabajo la técnica a utilizar para la limpieza de datos es la técnica de detección de valores anómalos debido a que es la más acorde a la investigación. En la investigación se trabaja con números que son resultado de cálculos para la radiación solar, por lo que tiende a existir datos anómalos que dificultan el resultado, es por ello que es necesario detectarlos para luego manejarlos de forma tal que no se elimine la información, sino que sea modificada.

PI7: ¿Qué impacto tendrá para Cuba el uso de algoritmos y métodos en la limpieza de datos para la toma de decisiones en el contexto de las fuentes de energías renovables?

Cuba tiene como objetivo priorizar el sector energético, potenciando la diversificación de la matriz energética con el uso de energías renovables. Las decisiones para invertir en ellas, resultan cada día más problemáticas debido a la complejidad actual de los sistemas socioeconómicos y a las particularidades del sector. Para una adecuada selección de la inversión a acometer, se deben tener en cuenta un conjunto de análisis. Estos análisis en la actualidad, se realizan de forma manual, por lo que esto trae consigo el aumento de datos erróneos y la disminución de una buena calidad en los datos.

Por otro, lado varios de los conjuntos de datos se encuentran en diferentes *dataset* y difieren en cuanto a formato, por lo que hacen que el proceso de toma de decisiones sea más complejo y largo. Para evitar todo lo anteriormente planteado, se deben llevar a cabo procesos de limpieza de los datos en este sector. Como resultado de estos, se reducen los costos para el empleo de recursos adicionales sin aumentar los niveles de personal. La limpieza de datos es una función empresarial necesaria que puede subcontratarse sin problemas, lo que ayuda a mantener una información crítica para el negocio con una alta calidad.

Conclusiones parciales.

El cálculo de la radiación solar para el desarrollo de la energía solar en Cuba requiere de una serie de atributos que aportan diferentes características para la implementación de nuevos recursos y tecnologías.

En el desarrollo del método de la revisión sistemática de la bibliografía se ha evidenciado que no existen en Cuba un proceso de revisión de la calidad de la información registrada de los datos en esta rama. Sin embargo, existen una serie de algoritmos para la detección de anomalías que se pueden utilizar para este proceso de verificación de la calidad.

Los algoritmos para la técnica de detección de anomalías en este sistema de información son recomendados debido a que genera una información más confiable y llevadera para el desarrollo de tecnologías como sistemas fotovoltaicos o calentadores solares.

No existen repositorios en la web de datos publicados referidos a energía solar en Cuba. En la revisión de la bibliografía que se realiza durante el desarrollo de la investigación, se evidencia que actualmente la publicación de los datos en la web se enfoca utilizando servicios.

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

2.1. Introducción

El presente capítulo está compuesto en dos secciones. En la primera sección se describe la propuesta de solución desde un enfoque teórico, basada en un método compuesto por siete fases para la detección de anomalías en *dataset* sobre radiación solar a partir de documentación sobre los mismos. En la segunda sección se describe la arquitectura, componentes, tecnologías y algoritmos aplicados en el desarrollo que implementa el método propuesto.

2.2. Método propuesto

Al modo estructurado y ordenado de obtener un resultado, descubrir la verdad y sistematizar los conocimientos se le conoce como método; al conjunto de instrucciones o secuencia de paso que se utiliza para realizar una tarea también es denominado método.

Para la realización de las fases se asume varios de los documentos que registran información sobre energías renovables. A partir de los ellos se tomó un *dataset* sobre *Cálculo de radiación solar* para la limpieza de datos y luego la visualización de los mismos mediante servicios web.

El método propuesto de limpieza de datos está compuesto por varias fases donde la salida de una fase constituye la entrada de la próxima. Las fases propuestas son:

1. Selección de fuentes de datos.
2. Pre-procesamiento.
3. Detección de anomalías
4. Manejo de anomalías.
5. Exportar datos.
6. Publicación de datos.
7. Visualización de los datos.

Para el desarrollo del método propuesto y teniendo como base la revisión sistemática del capítulo anterior se muestra en la siguiente figura la representación del mismo mediante un diagrama de flujo. En este diagrama se describe un orden lógico de los diferentes conceptos, algoritmos, artefactos y herramientas involucradas en cada una de las fases.

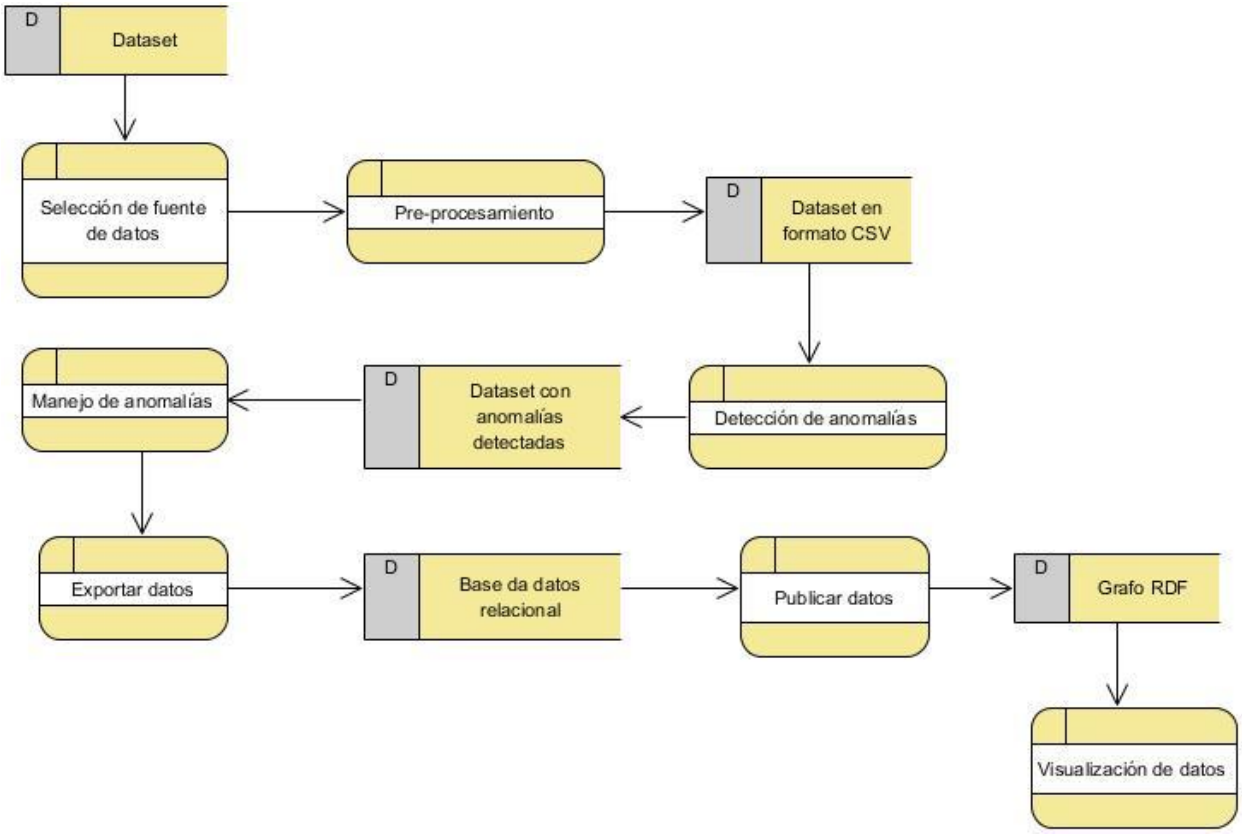


Figura 2 Método Propuesto

2.2.1 Selección de fuentes de datos

El Centro de Gestión de la Información y Desarrollo de Energía (CUBAENERGÍA) posee varios *dataset* en formato de hoja de cálculo (Excel). Para el desarrollo de la investigación se selecciona el que está relacionado con la intensidad de la luz, para el cálculo de radiación solar. Dicho *dataset* dispone de las siguientes mediciones:

Tabla III Descripción de las mediciones

Medición	Descripción
Global Horizontal	Radiación solar que incide sobre una superficie horizontal, representando así la suma de componentes como directa y difusa.
Global Normal	Radiación solar que incide sobre una superficie normal
Directa Normal	Radiación que llega directamente del foco solar.
45º Sur	Ángulo de inclinación de la superficie captadora
15º Sur	Ángulo de inclinación de la superficie captadora
Temperatura	Se analiza la medición de las temperaturas

Latitud	Valor de intensidad de la radiación solar según una latitud (no se hace referencia a que el valor sea punto geográfico)

El *dataset* describe la información para calcular la intensidad de la radiación solar sobre superficies con varias inclinaciones. Este contiene dos hojas, donde en la primera se muestran los resultados luego de llenar los campos latitud y longitud. La misma no es modificable para poder mantener la seguridad del *dataset*. Esta muestra los resultados a través de una tabla donde en la primera columna se encuentra el nombre de los datos que se van a mostrar como resultado, mientras que en las siguientes se encuentran los resultados obtenidos según los meses y una que tiene los datos anuales. Todos estos datos exceptuando *Temperatura* se dan en Kwh/m^2, mientras que la *Temperatura* se da en grados *Celsius*. Además, se muestra un gráfico que representa los niveles de radiación solar para cada parámetro, a través de líneas, solo los datos por meses del año. En el caso de la segunda hoja, contiene todas las mediciones obtenidas por los especialistas encargados de este tema, como latitud, longitud, distancia al punto 0, la temperatura y los datos según los parámetros por meses y anual (Ver Tabla II). Esta contiene un total de 19 200 entradas.

2.2.2 Pre-procesamiento

En esta fase se procura llevar todos estos datos a un único formato para que se puedan tratar más fácil con el fin de que sea más factible y cómodo, en este caso se llevan al formato *CSV*, mediante la biblioteca implementada en el lenguaje de programación Python, xlrd.

En el pre-procesamiento se realiza la extracción de la información y posterior transformación en tuplas. Una tupla matemáticamente es definida como una secuencia de valores agrupados. El tipo de datos que representa las tuplas se le llama tuple, este es inmutable debido a que ninguna tupla puede ser modificada una vez que se haya creado. En este caso las tuplas generadas se expresan de la siguiente manera:

$$Tupla = (\mathrm{L}, \mathrm{A}, \mathrm{T}, \mathrm{GH}, \mathrm{GN}, \mathrm{DN}, \mathrm{TP}, \mathrm{SC}, \mathrm{SQ})$$

Donde:

Tabla IV Descripción y definición de los parámetros.

Descripción	*Definición*
$\boldsymbol{L}$: $\boldsymbol{Longitud}$	$L = \{L \in \mathbb{R}\}$
$\boldsymbol{LG}$: $\boldsymbol{Longitud\ geográfica}$	$LG = \{LG \in \mathbb{R}\}$
$\boldsymbol{A}$: $\boldsymbol{Latitud}$	$A = \{A \in \mathbb{R}\}$
$\boldsymbol{T}$: $\boldsymbol{Type}$	$T = \{ANUAL, ENERO, FEBRERO, MARZO, ABRIL, MAYO, JUNIO, JULIO, AGOSTO, SEPTIEMBRE, OCTUBRE, NOVIEMBRE, DICIEMBRE\}$
$\boldsymbol{GH}$: $\boldsymbol{Global\ Horizontal}$	$GH = \{GH \in \mathbb{R}\}$

$\boldsymbol{GN}$: $\boldsymbol{Global\ Normal}$	$GN = \{GN \in \mathbb{R}\}$
$\boldsymbol{DN}$: $\boldsymbol{Directa\ Normal}$	$DN = \{DN \in \mathbb{R}\}$
$\boldsymbol{TP}$: $\boldsymbol{Temperatura}$	$TP = \{TP \in \mathbb{R}\}$
45°:	$SC = \{SC \in \mathbb{R}\}$
15°:	$SQ = \{SQ \in \mathbb{R}\}$

2.2.3 Detección de anomalías

Una anomalía es un dato muy distinto al resto, que se debe a fallos en mediciones, errores humanos o que no cumpla los criterios de calidad según la información registrada. A partir de tener como entrada el *dataset* en formato CSV generado en la fase anterior, se procede a aplicar la detección de anomalías, técnica de limpieza de datos. Existen varios algoritmos para realizar este proceso (ver sección 1.4 Resultados PI5 ***Algoritmos para la limpieza de datos***).

Para realizar la detección de anomalías se hace uso del algoritmo Z-score, el mismo permite la normalización de los datos en un conjunto de datos. Para comparar datos distintos, es necesario definir entonces, la fórmula a utilizar

$$(X - \mu)/\sigma$$

Donde:

- X: variable de observación, es la pequeña población que se observa y se estudia para la investigación y la detección de anomalías.
- μ: media general, se considera esta el valor verdadero.
- σ**:** variación estándar, esta mide la separación de los datos, o sea cuan separados están.

Una vez aplicada esta fórmula, el resultado de la misma se compara con un *threshold o umbral*, límite que se define para la comparación con el fin de deducir la existencia de la anomalía, este es otorgado para cada uno de los parámetros al que se le realiza la técnica (ver Tabla IV).

El threshold o umbral se define a partir de un criterio del experto, en este caso el especialista del Cubaenergía que se le realiza la entrevista.

Para detectar la anomalía el resultado de la diferencia de la observación puntual y la media dividida entre la variación estándar que se esté analizando es comparado con dicho threshold o umbral, en caso de que el resultado de esta observación sea mayor que dicho umbral, entonces se toma como anomalía final detectada en el algoritmo.

Tabla V Valor de threshold o umbral.

Parámetro a evaluar	Threshold
Latitud	4.6
Global Horizontal	2.9

Global Normal	4.5
Directa Normal	6.9
45°	5.3
15°	3.7
Temperatura	7.5

Otro algoritmo utilizado para la detección de anomalías es K-means. Este es un algoritmo de agrupamiento que se usa ampliamente en la minería de datos. Realiza particiones de *n* documentos en *k* agrupaciones donde cada objeto pertenece al agrupamiento con la media más cercana. A continuación, se describe la función objetivo que es utilizada por K-means.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2$$

Donde:

- J: es función objetivo
- k: número de clusters
- n: número de casos
- x: observación i
- c: centroide del cluster j

Encontrar una solución óptima para el agrupamiento de k-means es difícil de computar (NP-hard), sin embargo, existen heurísticas eficientes que se emplean para converger rápidamente a un óptimo local. La principal desventaja de la agrupación de k-means es que, de hecho, es muy sensible a la elección inicial del número de k. Por lo tanto, hay algunas técnicas utilizadas para determinar la k inicial, por ej. utilizando otro algoritmo de agrupamiento ligero como aglomerante(Allahyari, Trippe y Gutierrez 2017). Con el cálculo del centroide utilizando k-means se determina que para cada observación se aplique la siguiente fórmula:

$$|(X - c)| > \varepsilon$$

Donde:

- X: Observación
- c: centroide calculado
- ε: threshold o umbral

Para este algoritmo se hace uso de los umbrales definidos en la Tabla IV para cada de uno de los resultados del cálculo de la diferencia entre la observación puntual y el centroide. Aquí, al igual que en Z-Score, se aplica la comparación de dicho resultado con el umbral, donde si este resultado es mayor que el umbral es detectado finalmente una anomalía.

2.2.4 Manejo de anomalías.

Como resultado de la fase anterior se realiza la detección de anomalías. Según (Aguinis, Gottfredson y Joo 2013) una vez que las anomalías sean identificadas el procedimiento correcto es: o ajustar los datos a sus valores correctos o remover dichas observaciones del conjunto de datos. Para el manejo de anomalías se definen tres enfoques principales:

- Eliminación de anomalías: se basa en la eliminación de las anomalías una vez detectadas. Esta no es la opción recomendada debido a que se tiende a perder información que quizás resulta valiosa para el proceso.
- Marcado de anomalías: se basa en marcar aquellos valores anómalos, pero no se borran. Solo advierte de lo que pudiera ser eliminado para que no afecte así la información.
- Reescalado de anomalías: se enfoca en que los valores anómalos no tengan un efecto negativo, para ello se apoya en el uso de funciones matemáticas para reducir el impacto de la anomalía en los datos e información. Este enfoque se le aplica a todos los datos que se analizan, sin embargo no reescribe los datos originales sino que genera datos adicionales.

Para el desarrollo del método propuesto en esta fase se utiliza como enfoque el reescalado de anomalías, debido a que los valores que se tienen en el *dataset* son valores reales. En este enfoque se utiliza una función logarítmica para lograr el reescalado, debido a que tiende a disminuir la diferencia entre los datos. Dicha función está acotada solo para el caso del parámetro $Temperatura$ (ver Tabla III), debido a que los datos almacenados en el dataset, son datos del territorio cubano y en Cuba no hay temperaturas por debajo de 0. Por lo que para el uso de la función logarítmica el intervalo donde se debe acotar el parámetro anteriormente mencionado es $T \geq 2$. Utilizar esta función permite reducir la diferencia entre las observaciones normales y aquellas que presentan valores anómalos.

2.2.5 Exportar datos.

En esta fase los datos obtenidos a partir del manejo de anomalías son exportados hacia una base de datos relacional, donde se almacenan el nombre del *dataset* y los registros asociados al mismo, como se muestra en la *Figura 3.* En este esquema se tiene en cuenta solamente los datos asociados a la energía solar que son los analizados en el contexto de la investigación.

El repositorio de datos generados almacena cada uno de los valores asociados a las tuplas que se obtienen de las fases anteriores. Constituye la fuente de datos a utilizar para la tarea de publicación de datos.

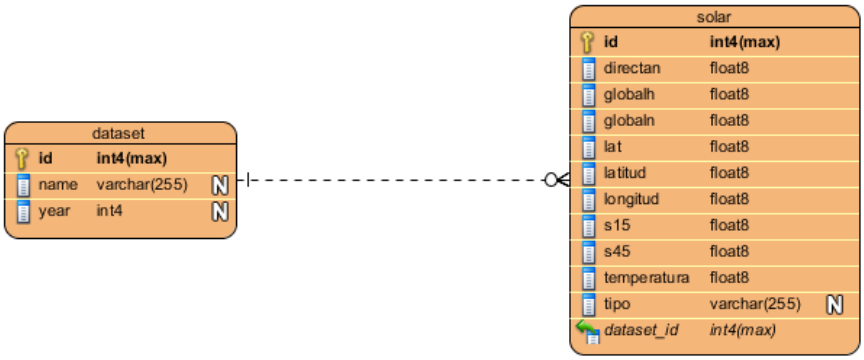


Figura 3 Diagrama Entidad-Relación de repositorio de datos.

2.2.6 Publicación de datos.

El objetivo de esta fase es publicar los datos obtenidos de las fases anteriores para su consumo a través de servicios y su publicación bajo estándares de la web semántica. Los datos que se publican en esta fase son los datos obtenidos a partir de las transformaciones del *dataset* original. Para el caso de los parámetros excepto $Temperatura$, los datos que son utilizados para la visualización se dividen entre un coeficiente con valor igual a 3.6. Dicho valor es definido por los especialistas y del cual se obtiene la radiación solar en Kwh/m^2.

2.2.6.1 Publicación como RDF

Utilizando el lenguaje de alineación de datos R2RML [6] para alinear los datos almacenados en el repositorio de datos con un grafo RDF utilizando un modelo de datos definido a priori para permitir la navegación a través de las tripletas RDF. El modelo de datos se estructura de acuerdo a las características deseadas para la publicación de los datos transformados en las fases anteriores.

2.2.6.2 Publicación como servicios REST

A partir de repositorio de datos se crean un conjunto de servicios REST que funcionan como proveedores de datos a aplicaciones de terceros que lo necesiten. Los datos publicados son datos estructurados, teniendo en cuenta las restricciones que se aplican durante la transformación de los mismos.

2.2.7 Visualización de datos

Utilizando los servicios web obtenidos en la etapa anterior se realiza la visualización de los datos en forma de tabla y gráfico. La visualización de los datos se basa en la transformación que se realiza a partir de la división por el coeficiente 3.6 que permite expresar las medidas en función de Kwh/m^2.

[6] https://www.w3.org/TR/r2rml/

2.3 Implementación del método propuesto

Una vez descrito el método en la sección anterior, se hace necesario llevar a un escenario real la materialización del mismo. Se desarrolla una implementación del método en un prototipo funcional. A continuación, se describe el modelo ontológico, la arquitectura, estándares, tecnologías y algoritmos que se aplican indistintamente en las diferentes fases definidas.

2.3.1 Metodología

El desarrollo de software debe estar guiado por una metodología de desarrollo. Su aplicación correcta es garantía de un producto de calidad. Las metodologías se dividen en dos grupos: ágiles y tradicionales. No existe una metodología universal para cada tipo de proyecto. Se define una metodología según las características del equipo de desarrollo, el dominio de aplicación, el tipo de contrato, la complejidad y la envergadura del proyecto.

Dada la necesidad de desarrollar la propuesta de solución en un breve período de tiempo, garantizando además la flexibilidad necesaria en cuanto a la variación de los requisitos y el manejo de los riesgos técnicos, así como reducir la generación de documentos y artefactos, y no habiendo un contrato tradicional, siendo el cliente parte del equipo de desarrollo; se hace necesario optar por un enfoque ágil de desarrollo de software en lugar de un enfoque tradicional o pesado. Teniendo en cuenta lo anterior han sido evaluadas varias metodologías que siguen el enfoque ágil de desarrollo de software, tal es caso de *Scrum* (Sutherland 2016), *Agile Unified Process* (AUP, por sus siglas en inglés) (Edeki 2013), *Extreme Programming*[7] (XP, por sus siglas en inglés) (Kniberg 2015) y AUP-UCI. En la investigación se adopta la variación de la metodología AUP para la UCI, específicamente en el escenario cuatro; puesto que se ajusta para proyectos que no modelan un negocio sino modelan el sistema con las Historias de Usuario (HU, por sus siglas en español), siendo éste el caso que ocupa.

2.3.2 Requisitos

La definición de requisito en la literatura científica cuenta con varias acepciones. La Real Academia Española[8] lo define como: circunstancia o condición necesaria para algo. Por otra parte, IEEE en (ISO 2018) enuncia el concepto de la siguiente manera: declaración que se traduce o expresa una necesidad y sus limitaciones y las condiciones correspondientes. A continuación, se listan las técnicas para su captura.

2.3.2.1 Técnicas de captura de requisitos

1. **Entrevista**

La entrevista es utilizada para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades. Según (Pressman 2010) , los miembros del equipo de software se reúne con

[7] http://www.extremeprogramming.org

[8] http://dle.rae.es/?id=W6xh4wt

los usuarios para entender mejor sus necesidades, motivaciones, cultura laboral y una multitud de aspectos adicionales.

Para la investigación se entrevista a un especialista a cargo del tema de energía solar, *Manuel Joaquín Álvarez González*, Licenciado en Física, Universidad de la Habana, 1976, Vicedirector de Energía del Centro de Gestión y Desarrollo de Energía (CUBAENERGIA) del Ministerio de Ciencia, Tecnología y Medio Ambiente de Ciudad de la Habana (CITMA).

Como resultado de la entrevista se obtuvo la descripción de la labor que se realiza para el desarrollo de la energía solar en Cuba, además de que se pudo tener acceso al *dataset* sobre la intensidad de la luz para el cálculo de la radiación solar en Cuba, *dataset* que es utilizado para el desarrollo de la investigación. Se definieron, en la entrevista, las reglas de negocio para el trabajo con los datos, en el proceso de extracción, limpieza y publicación de los datos de energía solar. *Ver Anexos Tablas*

2. Prototipado

Los prototipos suelen consistir en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida. Esta técnica es particularmente útil cuando:

- El área de la aplicación no está bien definida (posiblemente por ser algo muy novedoso).
- Es necesario evaluar previamente el impacto del sistema en los usuarios y en la organización.

Los prototipos de sistema permiten a los usuarios experimentar para ver cómo éste ayuda a su trabajo. Fomentan el desarrollo de ideas que se convierten en requerimientos.

3. Observación

Por medio de esta técnica el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades. Este método permite observar la forma en que se llevan a cabo los procesos y, por otro, verificar que realmente se sigan todos los pasos especificados. Como sabemos, en muchos casos los procesos son una cosa en papel y otra muy diferente en la práctica. Los observadores experimentados saben qué buscar y cómo evaluar la relevancia de lo que observan.

4. Estudio de documentación

Varios tipos de documentación, como manuales y reportes, pueden proporcionar al analista información valiosa con respecto a las organizaciones y a sus operaciones. La documentación difícilmente refleja la forma en que realmente se desarrollan las actividades, o donde se encuentra el poder de la toma de decisiones. Sin embargo, puede ser de gran importancia para introducir al analista al dominio de operación y el vocabulario que utiliza.

Como documentos, de la literatura especializada en el tema, estudiados se tiene el libro “Manual para el cálculo y diseño de calentadores solares”, cuyo autor es el especialista entrevistado ***(ver sección 2.3.2.1 Entrevista)***. Además, otro documento para el análisis lo constituye el dataset para la intensidad de la luz para el cálculo de la radiación solar y la tesis de maestría “Atlas de bioenergía. Cuba sector agropecuario y forestal”.

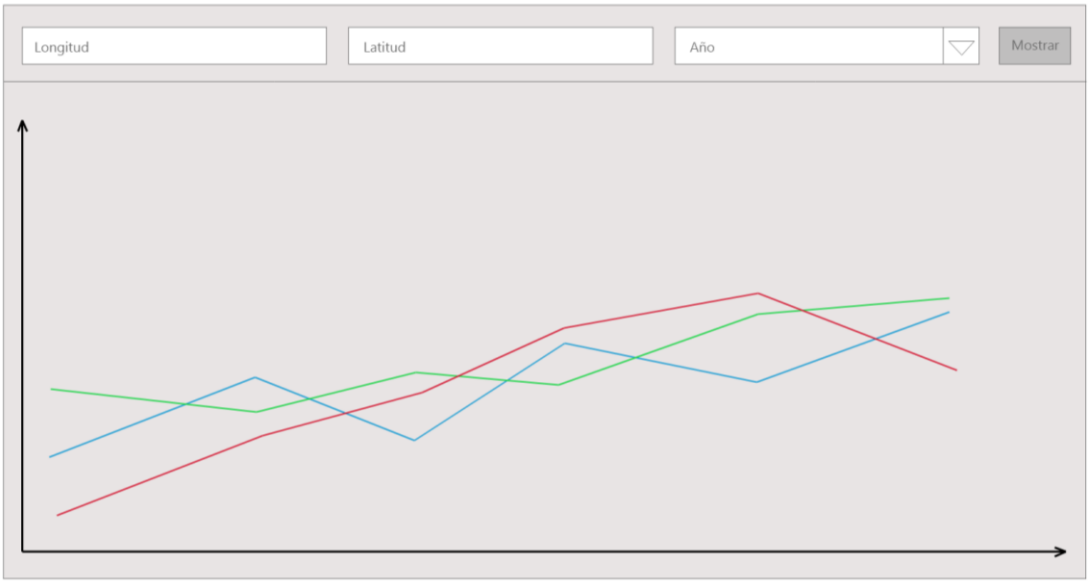
2.3.2.2 Requisitos funcionales

Las historias de usuario son utilizadas en las metodologías de desarrollo ágil puesto que son una forma rápida para la especificación y administración de requisitos, sin necesidad de elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes y deben cumplir varias restricciones, entre ellas: ser independientes unas de otras, negociables, estimables, pequeñas y verificables, por mencionar algunas. En la investigación se definen los requisitos funcionales de acuerdo a las etapas definidas en el método propuesto (ver sección **2.2 Método propuesto**), exceptuando la etapa "Selección de datos", obteniéndose un total de 6 requisitos funcionales. En la Tabla VI se muestra la historia de usuario para el requisito "Visualización de datos", para las restantes ver sección ***Anexos***.

Tabla VI Historia de usuario del requisito: Visualización de datos

*Código: **HU-06***	Nombre del requisito: **Visualización de datos**
Programador*:* *Tahimí González Oliva*	Iteración asignada:1
Prioridad*: alta*	Tiempo estimado: 2 semanas
Riesgo en desarrollo*: alto*	Tiempo real: 2 semana
Descripción*: Esta tarea tiene como objetivo proveer de una interfaz web para mostrar los datos del cálculo de radiación solar según longitud, latitud y año del dataset (ver sección **2.2.7 Visualización datos**).*	

Prototipo de Interfaz



Estimación de esfuerzo por historias de usuario.

Dependiendo de la prioridad asignada por el cliente a cada historia de usuario y atendiendo a la complejidad y riesgo determinado por el programador, se define la estimación de cada una de las historias de usuario identificadas. La Tabla X muestra los resultados de la estimación realizada. La unidad de estimación es el punto, y un punto equivale a una semana ideal de programación.

Tabla VII Puntos de estimación por historia de usuario

Historias de usuario	**Puntos de estimación**
Pre-procesamiento de datos	2
Detección de anomalías	4
Manejo de anomalías	2
Exportar datos	1
Publicar datos	2
Visualización de datos	2

2.3.3 Modelo ontológico

Un modelo ontológico brinda definiciones precisas sobre conceptos, relaciones de los mismos y reglas que rigen el dominio. La aplicación de un modelo ontológico permitirá la creación o reutilización de ontologías para la modelación de los datos en un grafo RDF.

Para la investigación se utilizó un grafo RDF que maneja las ontologías de la Tabla VIII.

Tabla VIII Descripción de las ontologías presentes en el grafo RDF.

Nombre	Fuente de la ontología	Descripción
rdfs	http://www.w3.org/2000/01/rdf-schema#	Esquema de vocabulario RDF.
fabio	http://purl.org/spar/fabio/	Es una ontología para el registro y publicación de registros bibliográficos en la Web Semántica de trabajos académicos.
foaf	http://xmlns.com/foaf/0.1/	Ontología que describe a las personas, sus actividades y sus relaciones con otras personas y objetos.
geo	http://www.w3.org/2003/01/geo/wgs84_pos#	Ontología que describe un vocabulario RDF básico para representar la longitud y latitud y otras informaciones sobre objetos espacio-localizados usando la referencia WGS84.

Utilizando coma base la ontología definida se utiliza un modelo de datos que permita realizar la generación de grafos RDF, ver Figura 4.

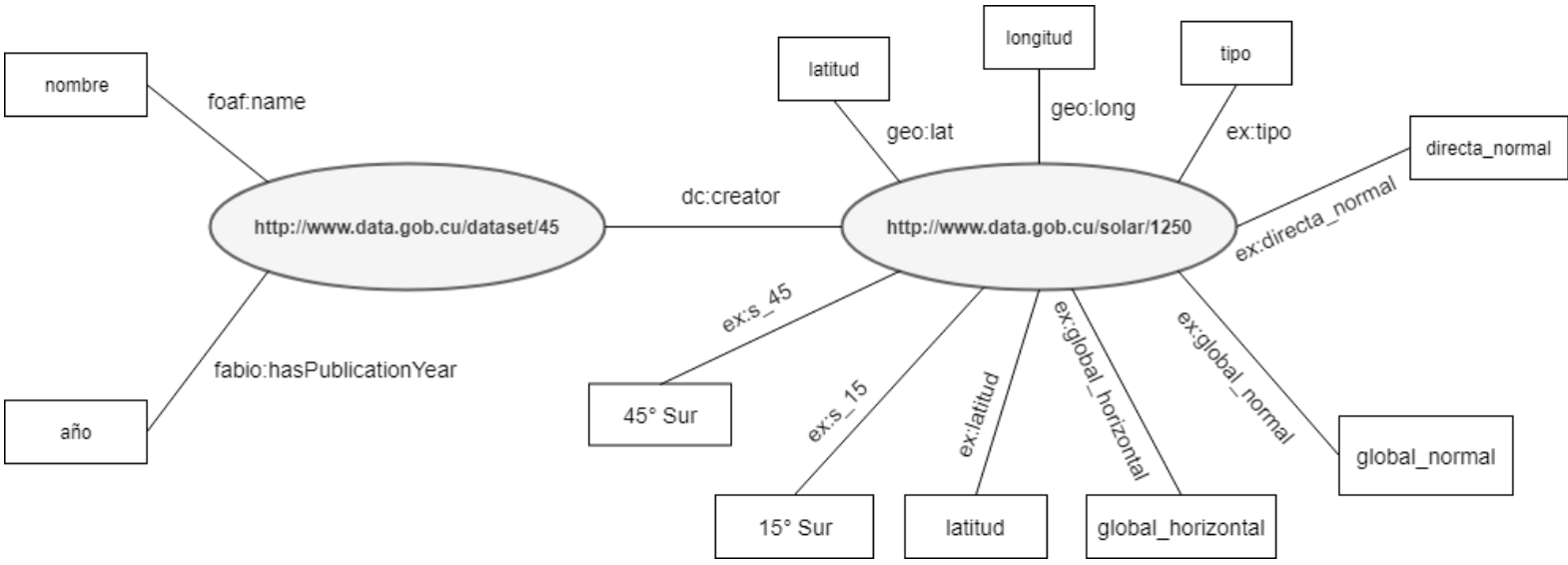


Figura 4 Modelo datos del grafo RDF

2.3.4 Arquitectura

La propuesta de solución presenta una arquitectura de flujo de datos. Esta arquitectura se aplica cuando datos de entrada van a transformarse en datos de salida a través de una serie de componentes computacionales o manipuladores (Pressman 2010). El estilo arquitectónico a utilizar es el de tuberías y filtros, debido a que este es un procedimiento de datos conectados en serie, donde la salida de un

elemento constituye la entrada del siguiente. Este patrón tiene un conjunto de componentes, llamados *filtros*, conectados por *tubos* que transmiten datos de un componente al siguiente(Pressman 2010).

La arquitectura obtenida se muestra en la Figura 5 donde se refleja el flujo de información y la interacción que se establece entre los módulos y componentes que se definen en la instanciación del modelo propuesto.

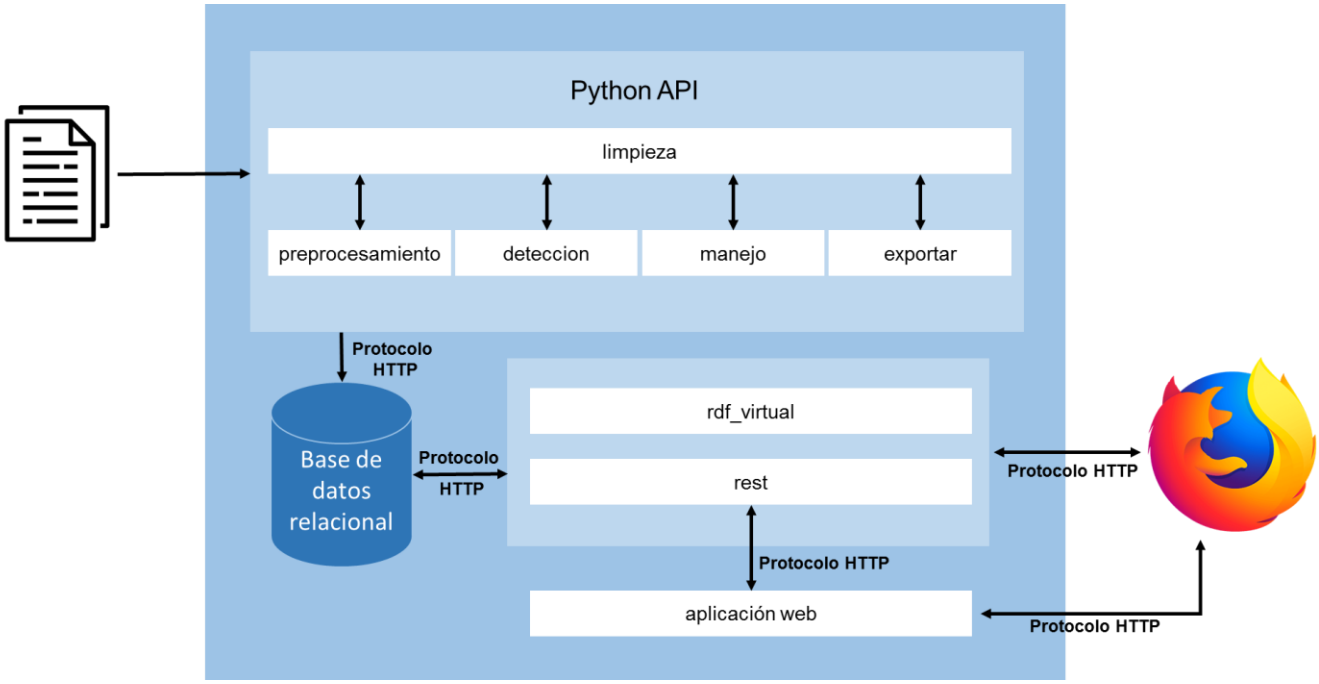


Figura 5 Arquitectura del método propuesto.

Los datos de entrada son hojas de cálculo que poseen información sobre la intensidad de la radiación solar teniendo en cuenta diferente métricas. Los componentes de la aplicación van procesando la información de acuerdo al estado en el que se encuentre. El módulo ***limpieza*** se encarga de controlar el flujo de los datos durante el proceso de extracción y limpieza hasta su posterior publicación en una base de datos relacional. En ***preprocesamiento*** se realiza, la extracción de los datos procedentes de los documentos de entrada. El pre-procesamiento de los datos se realiza mediante la librería XLRD implementada en el lenguaje de programación Python con el fin de llevarlos todos al formato CSV, formato estándar abierto de archivo de texto plano, más recomendado para datos tabulados. Los dataset generados en ***detección*** son analizados para la detección de anomalías en los mismos. Las métricas que posean datos anómalos son utilizadas por **manejo** para realizar el manejo de los datos que son anómalos en las métricas que los posean y se genera un archivo donde se almacenan los datos con las trasformaciones realizadas. El módulo ***exportar*** se encarga de transformar los datos analizados hacia un esquema de base de datos relacional.

Los datos almacenados en la base de datos relacional son utilizados para la publicación de los datos a través de servicios web. El servicio que se provee en **rest** es para el consumo de los datos generados utilizando el protocolo REST. Adicionalmente, en **rdf_virtual** se publica un SPARQL Endpoint, se permite la generación de ficheros RDF con los datos contenidos en la base de datos y se provee de navegación virtual como RDF a través de los datos almacenados.

limpieza: módulo principal que se encarga de controlar todo el flujo de información y las transformaciones que se realizan en las dos primeras fases del método propuesto.

preprocesamiento: módulo que se encarga de transformar los datos de la estructura original hacia la estructura deseada (ver sección 2.2.2).

detección: módulo que gestiona los algoritmos encargados de la detección de anomalías en el dataset que se genera a partir de los datos iniciales. Para la detección de las anomalías se aplican las restricciones definidas en el método propuesto (ver sección **2.2.3 Detección de anomalías**). La ejecución de los algoritmos se realiza de manera secuencial para evitar el problema de acceso a una misma sección crítica de memoria simultáneamente lo que pudiera provocar un malfuncionamiento de los algoritmos y obtener resultados no deseados.

manejo: módulo que se encarga de realizar el manejo de las anomalías que se detecten utilizando una función $\log$para el re-escalado de los datos que posean anomalías. Se ejecuta tantas veces como métricas con anomalías se hayan detectado.

exportar: módulo que se encarga de guardar los datos almacenados en el dataset que se genera en una base de datos relacional.

rdf_virtual: módulo que se encarga de proveer un SPARQL Endpoint para el consume de datos como datos enlazados. Permite generar RDF con serialización de formato en *Turtle*(ttl). Utilizando un esquema de alineación generado se encarga de la navegación a través de los datos de la base de datos como RDF virtual. La Figura 6, muestra un fragmento del grafo RDF obtenido, teniendo como desventaja la demora en el proceso de generación del grafo RDF para grandes volúmenes de datos, siendo este un problema abierto en la comunidad científica.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:fabio="http://purl.org/spar/fabio/"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:map="http://localhost:2020/#"
    xmlns:vocab="http://localhost:2020/vocab/"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:db="http://localhost:2020/"
  xml:base="http://localhost:2020/" >
  <rdf:Description rdf:about="solar/186666">
    <vocab:solar_dataset_id rdf:resource="dataset/11"/>
    <vocab:solar_s45 rdf:datatype="http://www.w3.org/2001/XMLSchema#double">20.216829999999998E0</vocab:solar_s45>
    <geo:long rdf:datatype="http://www.w3.org/2001/XMLSchema#double">-75.6E0</geo:long>
    <vocab:solar_globaln rdf:datatype="http://www.w3.org/2001/XMLSchema#double">23.56965E0</vocab:solar_globaln>
    <vocab:solar_directan rdf:datatype="http://www.w3.org/2001/XMLSchema#double">19.48616E0</vocab:solar_directan>
    <vocab:solar_id rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">186666</vocab:solar_id>
    <geo:lat rdf:datatype="http://www.w3.org/2001/XMLSchema#double">22.45E0</geo:lat>
    <vocab:solar_globalh rdf:datatype="http://www.w3.org/2001/XMLSchema#double">15.84702E0</vocab:solar_globalh>
    <vocab:solar_lat rdf:datatype="http://www.w3.org/2001/XMLSchema#double">19.02327E0</vocab:solar_lat>
    <rdfs:label>solar #186666</rdfs:label>
    <vocab:solar_temperatura rdf:datatype="http://www.w3.org/2001/XMLSchema#double">29.238709999999998E0
    </vocab:solar_temperatura>
    <vocab:solar_tipo>NOVIEMBRE</vocab:solar_tipo>
    <vocab:solar_s15 rdf:datatype="http://www.w3.org/2001/XMLSchema#double">18.29505E0</vocab:solar_s15>
    <rdf:type rdf:resource="vocab/solar"/>
  </rdf:Description>
```

Figura 6 Grafo RDF generado

rest: paquete encargado de publicar los datos para su consumo por aplicaciones de terceros utilizando el protocolo REST. La Tabla IX muestra los principales servicios web que se proveen a través de la API REST implementada.

Tabla IX Descripción de servicios API REST de publicación

Direcciones		Descripción
Get	GET /solar/api/dataset/all	Devuelve en formato Json todos los dataset publicados.
	GET /solar/api/dataset/test	Devuelve el estado de la API, si está en ejecución o no.
	GET /solar/api/dataset/years	Devuelve los años de publicación en formato Json de todos los dataset disponibles.

	GET /solar/api/solar/all	Devuelve todos los registros en format Json de los datos asociados a la energía solar.
Post	POST /solar/api/solar/coordenadas	Dado las coordenadas de latitud y longitud devuelve los valores de radiación específicos para las coordenadas dadas.
	POST /solar/api/solar/radiacion/{year}	Devuelve los valores de radiación solar pertenecientes al año seleccionado.

aplicación web: aplicación web para mostrar los datos publicados como servicios. Los datos son mostrados en forma gráfica en correspondencia con los valores de radiación solar sobre superficies de varias inclinaciones, estos datos están dado en Kwh/m^2 (ver **Anexo 1 - 3**).

Los patrones arquitectónicos y de diseño se consideran una serie de buenas prácticas basadas en la experiencia y se ha demostrado que funcionan en el desarrollo del software. Son soluciones a problemas específicos y comunes del diseño orientado a objeto. En el desarrollo de la solución propuesta fueron utilizados, entre otros, los patrones que se detallan a continuación:

- **Controlador (**en inglés ***Controller*****):** es un patrón de tipo *General Responsibility Assignment Software Patterns* (GRASP). Consiste en asignar la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. La solución posee varias clases controladoras, una de ellas controla el flujo normal de eventos y de la interacción entre los paquetes y las restantes clases controlan el flujo de las peticiones HTTP entre el navegador web y los servicios web publicados.
- **Alta cohesión:** es un patrón de tipo *General Responsibility Assignment Software Patterns* (GRASP), un principio evaluativo que aplica un diseñador mientras evalúa todas las decisiones diseño. Indica la relación que existe entre los elementos de un mismo módulo. Este patrón se ve reflejado en ***limpieza***, pues hace función de módulo y gestiona el flujo de información, durante la extracción y limpieza de los datos.
- **Creador:** es un patrón de tipo *General Responsibility Assignment Software Patterns* (GRASP), guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. Este patrón se ve reflejado ***SolarDataController****,* esta crea instancias de la clase ***SolarDataRepository.***
- **Bajo acoplamiento:** es un patrón de tipo *General Responsibility Assignment Software Patterns* (GRASP), impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que conlleve a resultados negativos que puede

producir un acoplamiento alto. Es el grado de interdependencia entre los módulos. En la propuesta de solución se ve reflejado con la clase ***preprocesamiento*** que permite la transformación de los datos.

- **Experto:** es un patrón de tipo *General Responsibility Assignment Software Patterns* (GRASP), este patrón se evidencia en la propuesta de solución con la clase ***SolarData,*** esta clase permite el encapsulamiento de la información relacionada con los usuarios del sistema.

2.3.5 Bibliotecas

Pandas

Paquete de Python que proporciona estructuras de datos rápidos, flexibles y expresivos diseñados para hacer que el trabajo con datos estructurados (tabulares, multidimensionales, potencialmente heterogéneos) y de series de tiempo sea fácil e intuitivo. Además, tiene el objetivo de convertirse en la herramienta de análisis/manipulación de datos de código abierto. *Pandas* es adecuado para muchos tipos de datos:

- Datos tabulares con columnas de tipo heterogéneo, como tablas de SQL u hojas de cálculo.
- Datos de series de tiempo ordenados y desordenados (no necesariamente de frecuencia fija).
- Datos matriciales arbitrarios (tipificados homogéneamente o heterogéneos) con etiquetas de fila y columna
- Cualquier otra forma de conjuntos de datos observacionales/estadísticos. En realidad, no es necesario etiquetar los datos para colocarlos en una estructura de datos de *pandas.* (Python Software Fundation 2019a)

XLRD

Biblioteca Python para extraer datos de hojas de cálculo de Excel (.xls y .xlsx, versiones 2.0 en adelante) (Python Software Fundation 2019b)

NumPy

NumPy es el paquete fundamental para la computación científica con Python. Contiene entre otras cosas:

- un objeto de matriz N-dimensional
- Funciones sofisticadas
- Herramientas para la integración de código C / C ++ y Fortran.
- Álgebra lineal útil, transformada de Fourier y capacidades de números aleatorios.

Además de sus usos científicos, *NumPy* también puede usarse como un eficiente contenedor multidimensional de datos genéricos. Se pueden definir tipos de datos arbitrarios. Esto permite que *NumPy* se integre con una amplia variedad de bases de datos(NumPy Developer 2019).

Highcharts

Motor de gráficas JavaScript más popular del mundo. Posee integración con HTML, compatible con navegadores modernos que incluyen dispositivos móviles, tabletas e IE antiguo hasta IE6. Genera graficas de diferentes tipos de manera dinámica(Highsoft 2019).

React JS

Biblioteca de JavaScript para construir interfaces de usuario. Utiliza una sintaxis similar a XML llamada JSX. Se basa en el uso de componentes que poseen estados. Los estados almacenan la información de los componentes. React permite interactuar con otras bibliotecas y frameworks (Facebook Open Group 2019).

2.3.6 Estándares y Tecnologías

RDF (Resource Description Framework): modelo de datos que utiliza XML, Ntriples, Turtle, entre otros, como lenguaje para serializar los datos y metadatos de los recursos de la web. Dicho modelo de datos constituye la fuente principal de información que almacena los metadatos bibliográficos necesarios para el desarrollo del prototipo funcional. Permite la intemporalidad entre las aplicaciones que intercambian información comprensible por la página Web, para proporcionar una infraestructura que soporte actividades de metadatos. Posibilita la utilización de cualquier grado RDF que almacene metadatos bibliográficos como fuentes de datos del prototipo funcional, debido que es una estándar a nivel mundial.

API REST (*Application Programming Interface) o Interfaz de Programación de Aplicaciones:* permite a los desarrolladores que interactúen con los datos de la aplicación de un modo planificado y ordenado. Por otro lado, la API *Rest* es utilizada para crear servicios, debido a que es un estándar lógico y eficiente para la creación de servicios web. Las operaciones más importantes que permite manipulas la API *Rest* son: GET para consultar y leer, POST para crear, PUT para editar y DELETE para eliminar. El uso de un conjunto de procedimientos para crear contenidos que contengan textos, imágenes, vídeo, audio entre otros métodos de la información que permite al usuario navegar por distintos recursos de una API *Rest* se realiza a través de enlaces HTML. Las API *Rest* es mucho más efectivo gracias a HTTP *(Hyper Text Transfer Protocol).* El motivo de que esto sea así es que este protocolo permite compartir información entre un cliente (portátil, teléfono entre otros) y un servidor. Alguno de los framework para la implementación de las APIs más usados son: JAX-RS y Spring Boot para Java, Django REST framework para Python, Laravel para PHP.

Node JS[9]**:** es un entorno de ejecución para JavaScript orientado a eventos asíncronos, está diseñado para construir aplicaciones en red escalables. Node lleva el modelo de eventos un poco más allá, este

[9] https://nodejs.org/

presenta un bucle de eventos como un entorno en vez de una librería. Diseñado con operaciones de *streaming* y baja latencia. Esto hace que Node sea candidato para ser la base de una librería o un *framework* web.

Spring Boot: marco de código abierto basado en Java que se utiliza para crear un microservicio. Es desarrollo por *Privotal Team* y se usa para construir aplicaciones *spring* independientes y listas para la producción. Proporciona una plataforma para que los desarrolladores de Java puedan construir una aplicación de *spring*. Puede comenzar con las configuraciones mínimas sin la necesidad de una configuración completa de la configuración de Spring. Tiene como ventaja la facilidad de entender y desarrollar aplicaciones de *spring*, aumenta la productividad y reduce el tiempo de desarrollo.

Tomcat[10]**:** el servidor Tomcat es una extensión a la tecnología Java para la ejecución de *scrips* en servicios Web. Proporciona implementaciones de código abierto de bibliotecas de etiquetas para su uso con páginas de servidor Java. En particular, aloja la etiqueta estándar de Apache, una implementación de código abierto de la especificación de la biblioteca de etiquetas estándar de Java.

Boostrap: es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

2.3.7 Lenguaje de programación

Python 3.7: lenguaje simple por su sintaxis por lo que es más fácil de leer para el programador. Presenta un tipado dinámico siendo esto una posibilidad que se le brinda al usuario para cambiar el tipo de variable sin tener que declararla. Actualmente domina el mundo de la Inteligencia Artificial, la ciencia de datos, entre otros. Python dispone de librerías y gran cantidad de módulos. Destacado por ser multi-paradigma, esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos**,** programación imperativa y programación funcional. En el caso específico de Python 3.7, es una versión, que se creó con el fin de mejorar la optimización de los programas y la creación de nuevas funciones.

Java 1.8: lenguaje de programación orientado a objetos con muchas bibliotecas útiles. Permite desarrollar y desplegar aplicaciones Java en ordenadores personales y servidores, así como en los exigentes entornos integrados. Java ofrece la interfaz del usuario, rendimiento, versatilidad, portabilidad y seguridad que las aplicaciones actuales requieren. A partir de la versión 8 se incluyen los conceptos

[10] http://tomcat.apache.org

de *stream* y funciones dentro del lenguaje(Urma, Fusco y Mycroft 2015). Durante el desarrollo de utiliza como base para el desarrollo de la API REST.

2.3.8 Herramientas

Postman: herramienta dirigida a desarrolladores web que ayuda en el desarrollo y prueba APIs, permitiendo hacer peticiones a la misma, creando distintos escenarios de pruebas. Ofrece la posibilidad de comprobar el correcto funcionamiento del desarrollo. Postman presenta como características principales la documentación sobre API detallada y visualizable en la web, además del monitoreo de API flexible para disponibilidad, rendimiento y precisión. Por otro lado, tiene servidores simulados para soportar el desarrollo. Proporciona todas las herramientas que se necesita para la administración de cada una de las etapas del ciclo de vida de la API.

Adobe XD: herramienta desarrollada por Adobe Inc. orientada al diseño de prototipos de interfaz de usuario para diferentes plataformas (web y móvil). Permite realizar diseños interactivos de flujos de una aplicación.

PyCharm: entorno de desarrollo integrado (IDE) utilizado en la programación, específicamente en el lenguaje de programación Python. Es un IDE multiplataforma, con versiones de Windows, macOS y Linux. PyCharm es uno de los entornos de desarrollo más completo. Trabajar con este IDE tiene como ventaja la posibilidad de refactorizar el código, lo que significa modificar el código sin comprometer la ejecución del mismo. Presenta una integración con frameworks web como: Django, Flask, Pyramid, Web2Py y por otro lado frameworks javascripts como: jQuery, AngularJS.

IntelliJ: entorno de desarrollo integrado para el desarrollo de programas informáticos posee las siguientes características(IDEA 2019):

- **Smart completion:** proporciona una lista de los símbolos más relevantes aplicables en el contexto actual.
- **Data flow analysis:** analiza el flujo de datos para estimar el posible tipo de símbolo de tiempo de ejecución y refina las opciones basadas en esa información, agregando automáticamente opciones de clase.
- **Cross-language refactorings:** provee refactorizaciones efectivas y completas. Por ejemplo, cuando Renombra una clase dentro de una declaración JPA[11], se actualizará todo, desde la clase de entidad JPA, a cada expresión JPA donde se usa.

[11] Java Persistence Application

En el desarrollo de la propuesta de solución se utiliza para el desarrollo de la API Rest para proveer de un servicio web para el consumo de los datos generados.

Visual Paradigm 8.0: *visual paradigm* para UML *Standard Edition* es una plataforma de modelado diseñado para ayudar a los arquitectos de sistemas, desarrolladores y diseñadores de UML a acelerar el proceso de análisis y diseño para aplicaciones empresariales complejas. Es una tecnología de modelado visual que facilita la visualización de UML en diferentes tipos de diagramas.

PostgreSQL (9.4): es un sistema de base de datos relacional de objeto de código abierto de desarrollo de alta confiabilidad, integridad de datos, solidez de funciones y rendimiento. Utiliza y amplia el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajos de datos más complicadas. Además, PostgreSQL es altamente extensible, muchas características, como los índices, tienen API definidas para que pueda desarrollarse con PostgreSQL para resolver sus desafíos. Es altamente escalable tanto por la gran cantidad de datos que puede administrar como por la cantidad de usuarios concurrentes que puede acomodar.

2.4 Conclusiones parciales

El método de solución para la limpieza y publicación de datos a partir de documentos de energía solar se basa en siete fases, siguiendo un enfoque basado en filtros y tuberías. Se detectan datos anómalos a partir del método que incluye algoritmos de limpieza de datos, solucionándose los problemas de detección de anomalías con diferentes criterios de medidas.

La arquitectura seleccionada a partir del método propuesto define el desarrollo del prototipo funcional, agrupándose las funcionalidades en componentes y evidencia el flujo de información planteado en las restricciones del método. Los estándares y tecnologías constituyeron en la investigación de los pilares fundamentales en el modelado de la solución e implementación del prototipo funcional. Lográndose transformar las restricciones definidas en el método propuesto a las correspondientes funcionalidades del prototipo funcional.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

3.1. Introducción

En este capítulo se realiza la validación para la detección de anomalías a partir de una hoja de cálculo que contiene datos de energía solar, aplicando un caso de estudio. El capítulo posee una única sección donde se describe paso a paso el desarrollo de las fases de la propuesta de solución para posteriormente realizar el análisis de los resultados obtenidos.

3.2. Pruebas de software

El proceso de pruebas de AUP constituye una de sus fortalezas, puesto que permite aumentar la calidad del sistema reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. En la variación de la metodología AUP para la UCI este proceso se desagrega en tres disciplinas: (1) pruebas internas, (2) pruebas de liberación y por último (3) pruebas de aceptación. En el caso específico de la investigación solo se tratan las pruebas internas.

3.2.1 Caja blanca

La prueba de caja blanca también es conocida como prueba de caja transparente o de cristal. Esta prueba consiste específicamente en el diseño de los casos de prueba atendiendo al comportamiento interno y la estructura del programa, examinándose la lógica interna sin tener en cuenta los aspectos de rendimiento. En la prueba de caja blanca se incluyen las técnicas descritas a continuación:

- ✓ Prueba del Camino Básico: permite obtener una medida de la complejidad lógica de un diseño y usar la misma como guía para la definición de un conjunto de caminos básicos.
- ✓ Prueba de Condición: ejercita las condiciones lógicas contenidas en el módulo de un programa. Garantiza la ejecución por lo menos una vez de todos los caminos independientes de cada módulo, programa o método.
- ✓ Prueba de Flujo de Datos: se selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Garantiza que se ejerciten las estructuras internas de datos para asegurar su validez.
- ✓ Prueba de Bucles: se centra exclusivamente en la validez de las construcciones de bucles. Garantiza la ejecución de todos los bucles en sus límites operacionales.

La técnica de caja blanca empleada en la solución propuesta fue la Prueba del Camino Básico, esta permite obtener una medida de la complejidad lógica del diseño procedimental y así usarla como guía para la definición de un conjunto básico de caminos de ejecución, garantizando con estos que durante la prueba se ejecute por lo menos una vez cada sentencia del método.

Para realizar la técnica anteriormente descrita es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar. Para ello se selecciona el método z_score_detection

(data, thresh, type_value, type_outliers), el cual retorna los outliers o anomalías detectadas en el dataset sobre energía solar.

A continuación, se enumera las sentencias de código del procesamiento realizados sobre el método z_score_detection (data, thresh, type_value, type_outliers), (ver Figura 7), y el grafo del flujo asociado al mismo, (ver Figura).

```
def z_score_detection(data, thresh, type_value, type_outliers):

    threshold = thresh                          #1
    mean_1 = np.mean(data[type_value])
    std_1 = np.std(data[type_value])
    outliers = []
    entry = {}
    index_val = 0
    for y in data[type_value]:                  #2
        z_score = (y - mean_1) / std_1          #3
        if np.abs(z_score) > threshold:         #4
            entry["index"] = index_val          #5
            entry["value"] = y
            outliers.append(entry)
            entry = {}

        index_val = index_val + 1               #6
    type_outliers[type_value]=outliers          #7
```

Figura 7 Código del método z_score_detection (data, thresh, type_value, type_outliers)

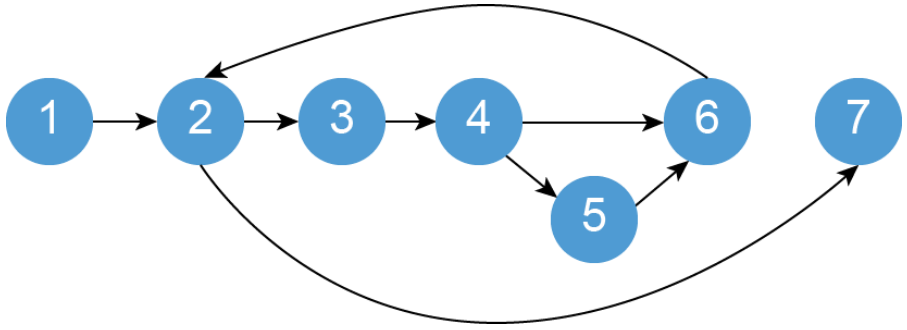


Figura 8 Grafo de flujo asociado al método z_score_detection (data, thresh, type_value, type_outliers)

Fórmulas para calcular la complejidad ciclomática

- ✓ V (G) = (A-N) + 2

  V (G) = (8-7) + 2

  V (G) = 3

Donde “A” es la cantidad de aristas y “N” la cantidad de nodos.

- ✓ V (G) = P + 1

  V (G) = 2+ 1

  V (G) = 3

Siendo “P” la cantidad de nodos predicados (son los nodos de los cuales parten dos o más aristas).

- ✓ V (G) = R

  V (G) = 3

Donde “R” representa la cantidad de regiones en el grafo.

El cálculo en cada una de las fórmulas aplicadas ha dado el mismo valor, por lo que se puede decir que la complejidad ciclomática del código es de 3, lo que significa que existen 3 posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Tabla X Caminos básicos del flujo

Número	**Caminos Básicos**
1	1-2-7
2	1-2-3-4-6-2-7
3	1-2-3-4-5-6-2-7

Se determinan los caminos básicos. Luego, se procede a ejecutar los casos de pruebas para cada uno de estos. Para definir los casos de pruebas es necesario tener en cuenta:

- ✓ Descripción: se describe el caso de prueba y forma general se tratan los aspectos fundamentales de los datos de entrada.
- ✓ Condición de ejecución: se especifica cada parámetro para que cumpla una condición deseada así ver el funcionamiento del procedimiento.
- ✓ Entrada: se muestran los parámetros que serán la entrada al procedimiento.
- ✓ Resultados esperados: se expone el resultado esperado que debe devolver el procedimiento después de efectuado el caso de prueba.

Tabla XI Caso de prueba de caja blanca para el camino básico # 1 (CPCB-01)

CPCB-01	
Descripción	Los datos de entrada para *data* en el *type_value* dado es 0.
Condición de ejecución	Al ser igual a 0 no entra al ciclo, por lo que no se le da valor a *type_outliers.*
Entrada	*data, thresh, type_value, type_outliers*
Resultados esperados	No se agrega ninguna anomalía nueva.
Evaluación de la prueba: Satisfactoria.	

Tabla XII Caso de prueba de caja blanca para el camino básico # 2 (CPCB-02)

CPCB-02	
Descripción	Los datos de entrada para *data* en el *type_value* dado es diferente de 0.
Condición de ejecución	Dentro de la estructura condicional no se cumple la condición.
Entrada	*data, thresh, type_value, type_outliers*
Resultados esperados	No se agrega ninguna anomalía nueva.
Evaluación de la prueba: Satisfactoria.	

Tabla XIII Caso de prueba de caja blanca para el camino básico # 3 (CPCB-03)

CPCB-01	
Descripción	Los datos de entrada para data en el *type_value* dado es diferente de 0.
Condición de ejecución	Dentro de la estructura condicional se cumple la condición.
Entrada	*data, thresh, type_value, type_outliers*
Resultados esperados	Se agregan anomalías nuevas.
Evaluación de la prueba: Satisfactoria.	

3.2.2 Test –Driven Development

El desarrollo dirigido por pruebas (TDD, por sus siglas en inglés) es una práctica recomendada en escenarios de desarrollo ágil de software. Codificación conocida como *test-first testing* o diseño impulsado por prueba, es una práctica en la que se le indica a un programador que escriba el código de producción solo después de escribir un caso de prueba automatizado que falla. Este enfoque ofrece una visión completamente opuesta al enfoque tradicional de *test-last* comúnmente utilizado en el desarrollo de software, donde el código de producción se escribe de acuerdo con las especificaciones de diseño, y, típicamente, solo después de que se escriba gran parte del código de producción, se debe escribir un código de prueba (Hammond y Umphress 2015).

Utilizando este enfoque se aplican dos tipos de pruebas: (1) unitarias y (2) de integración. En el caso de la primera es un nivel de prueba de software donde se prueban unidades/ componentes individuales de un software. El propósito es validar que cada unidad del software funciona según lo diseñado. Una unidad es la parte comprobable más pequeña de cualquier software (MYERS, BADGETT y SANDLER 2012). En la figura 9 y 10 se muestran dos métodos utilizados para el desarrollo de pruebas unitarias.

Las pruebas de integración de componentes de software se realizan después de las pruebas unitarias. El objetivo principal de las pruebas de integración es verificar el funcionamiento correcto de los módulos integrados. Los componentes integrados ya están probados en pruebas unitarias. El objetivo de las pruebas de integración es validar el rendimiento y la confiabilidad de los componentes del software (Sumra 2018).

```
def test_util_get_threshold_by_type(self):
    self.assertEqual(ut.get_threshold_by_type("DirectaN"),6.9)
    self.assertEqual(ut.get_threshold_by_type("Latitud"),4.6)
    self.assertEqual(ut.get_threshold_by_type("GlobalN"),4.5)
    self.assertEqual(ut.get_threshold_by_type("GlobalH"),2.9)
    self.assertEqual(ut.get_threshold_by_type("15S"),3.7)
    self.assertEqual(ut.get_threshold_by_type("45S"),5.3)
    self.assertNotEqual(ut.get_threshold_by_type("Temperatura"),7.4)# 7.5
```

Figura 9 Método de prueba que verifica los valores de umbral para los parámetros que se miden.

```
def test_outlier_detection(self):
    root = ut.get_project_root()
    temp = os.path.join(root, "temp/radiacion_solar-2018.csv")
    pf = pd.read_csv(temp)
    outliers_z_score={}
    outliers_k_means = {}
    z_score_detection(pf, 7.5, "Temperatura", outliers_z_score)
    kmeans_detection(pf, 7.5, "Temperatura", outliers_k_means)
    self.assertNotEqual(outliers_z_score, outliers_k_means)
```

Figura 10 Método de prueba que verifique los resultados obtenidos de aplicar los algoritmos para la detección de anomalías.

Las pruebas de integración tienes los siguientes objetivos (Sumra 2018):

- Comprobar el funcionamiento de los componentes integrados.
- Descubrir fallas o errores que no se detectan en las pruebas unitarias de los componentes.
- Conformar el riesgo de fallo del sistema.

En la Figura 11 se muestra una prueba de integración de un total de 5 realizada a la API Rest en la fase de publicación de datos.

```
60      @Test
61      public void whenValidInput_whenGetSolarData_thenStatus200() throws IOException, Exception {
62          Dataset dataset = new Dataset();
63          dataset.setName("Radiacion Solar");
64          dataset.setYear(2018);
65          datasetRepository.save(dataset);
66          ...
238         Dataset dataset1 = datasetRepository.findFirstByYear(2017);
239         CoordenadasForm coordenadasForm = new CoordenadasForm();
240         coordenadasForm.setLatitud(21.3);
241         coordenadasForm.setLongitud(-80.2);
242         mvc.perform(post( urlTemplate: "/api/solar/coordenadas")
243                 .contentType(MediaType.APPLICATION_JSON)
244                 .content(JsonUtil.toJson(coordenadasForm)))
245                 .andDo(print())
246                 .andExpect(status().isOk())
247                 .andExpect(content().contentTypeCompatibleWith(MediaType.APPLICATION_JSON))
248                 .andExpect(jsonPath( expression: "$", hasSize(greaterThanOrEqualTo( value: 2))));
249     }
```

Figura 11 Prueba de integración sobre API Rest en fase de publicación

Caso de estudio

La calidad de los datos es un concepto que varía en dependencia del contexto donde se aplique. Según (Pipino, Lee y Wang 2002) durante el análisis de la calidad de datos se definen dos formas principales de evaluación: (1) subjetiva y (2) objetiva. La subjetiva se enfoca en los individuos que trabajan con los datos a analizar, por otra parte, la objetiva se basa en el *dataset* que se va a utilizar. La evaluación objetiva se divide en dos métricas: (1) tarea-dependiente y (2) tarea-independiente. Las métricas definidas para tarea-dependiente, en las que se incluyen reglas de negocio de la organización, compañía o regulaciones gubernamentales, son utilizadas en contextos de específicos que las requieran. En el caso de la evaluación objetiva con tarea-independiente las métricas reflejan el estado de los datos sin conocimiento del contexto de aplicación que los genera. En el caso de la investigación se aplica las métricas definidas para este tipo de evaluación, junto a métricas subjetivas.

Las dimensiones o métricas a evaluar se utilizan en base a los análisis que realizan los investigadores durante el proceso de evaluación de los datos. En este contexto las principales dimensiones que son definidas dentro de la comunidad científica se describen en la Tabla XI.

Tabla XIV Dimensiones para calidad de datos

Dimensión	**Descripción**
Completitud	Grado en el cual no existen datos perdidos
Consistencia	Grado en el cual los datos son presentados en el mismo formato.

Unicidad	Grado en el cual los datos son únicos.
Validez	Grado en el cual los datos cumplen las reglas asociadas a los mismos.
Exactitud	Grado en el cual los datos representan el dataset.
Free-of-Error	Grado en el cual los datos son correctos y de confianza.
Reusabilidad	Grado en el cual los datos son simples, relevantes, accesibles y mantenidos en el tiempo.
Confianza	Grado en el cual se miden si los datos son confiables (Ej.: provienen de datos de gobiernos u organizaciones gubernamentales).

En la evaluación de las métricas se utiliza el caso de estudio como forma de validar la calidad de los datos. El estudio de casos permite analizar el fenómeno objeto de estudio en su contexto real, utilizando múltiples fuentes de evidencia, cuantitativas y/o cualitativas simultáneamente (Villarreal Larrinaga, O.;Landeta Rodríguez 2010).

La evaluación de las métricas se realiza sobre un *dataset* generado con 249 600 tuplas (ver sección **2.2.2 Pre-procesamiento**). Las evaluaciones en las que se utilizan procesos automatizados se realizan sobre un equipo de cómputo con las siguientes características de hardware: Procesador AMD A6-7310 4GB de RAM. En (Group 2013) se definen un conjunto de parámetros para la evaluación de las métricas para la calidad de los datos. En la investigación se describen las siguientes:

- Título: refiere el nombre de la métrica a evaluar.
- Definición: Como se define la métrica.
- Referencia: Reglas sobre cual dimensión va a ser comparada.
- Medida: Se establece el grado en el cual la dimensión se ajusta a su unidad de medida.
- Ámbito: Donde la dimensión es aplicable.
- Unidad de medida: Unidad en que se mide la dimensión evaluada.
- Tipo de medida: Puede tomar los siguientes valores:
    - Continua: Se verifica periódicamente, el cual cambia en el tiempo.
    - Discreta: Se verifica si una medida absoluta es verdadero o falso.
    - Evaluación: Se verifica utilizando algoritmos.
- Dimensiones relacionadas: Dimensiones asociadas.

A continuación, se describen cada una de las características anteriores para cada una de las métricas definidas. Las referencias (reglas de negocio) se describen en la Tabla XX.

Tabla XV Medición de la métrica Completitud

Título	**Completitud**
Definición	Proporción entre los datos almacenados y el potencial “100% completo“
Referencia	R1
Medida	Una medida de la ausencia de valores en blanco (nulo o cadena vacía) o la presencia de valores que no están en blanco.
Ámbito	0-100% de las tuplas.
Unidad de Medida	Por ciento
Tipo de medida	Evaluación
Dimensiones relacionadas	Validez y Exactitud

Tabla XVI Medición de la métrica Consistencia

Título	**Consistencia**
Definición	La ausencia de diferencia, cuando compara dos o más representaciones de una tupla.
Referencia	R2
Medida	Análisis de patrones y/o valor de frecuencia.
Ámbito	Evaluar las tuplas entre múltiples dataset
Unidad de Medida	Por ciento
Tipo de medida	Evaluación y Discreta
Dimensiones relacionadas	Validez y Exactitud

Tabla XVII Medición de la métrica Unicidad

Título	**Unicidad**
Definición	Ninguna tupla debe ser almacenada más de una vez
Referencia	R2
Medida	Análisis de las tuplas se realiza con las tuplas del propio dataset.
Ámbito	Evaluar las tuplas dentro del mismo dataset.
Unidad de Medida	Por ciento
Tipo de medida	Evaluación y Discreta
Dimensiones relacionadas	Consistencia

Tabla XVIII Medición de la métrica Validez

Título	**Validez**
Definición	Los datos son válidos si se estructuran utilizando la sintaxis definida.
Referencia	R3
Medida	Comparar los datos de origen con las tuplas generadas en el dataset
Ámbito	Todas la tuplas.
Unidad de Medida	Por ciento de elementos de datos que se consideran válidos a no válidos.
Tipo de medida	Evaluación, Continua y Discreta
Dimensiones relacionadas	Consistencia, Completitud, Unicidad y Exactitud

Tabla XIX Medición de la métrica Exactitud

Título	**Exactitud**
Definición	Grado en el cual los datos describen las mediciones que representan.
Referencia	R4
Medida	Grado con el cual los datos son capaces de reflejar las características de las mediciones realizadas
Ámbito	Todas la tuplas.
Unidad de Medida	Por ciento de elementos de tuplas que cumplen con las reglas de exactitud
Tipo de medida	Evaluación, Discreta
Dimensiones relacionadas	Validez

Tabla XX Medición de la métrica Libre-de-Error

Título	**Libre-de-Error**
Definición	Cantidad de datos que no presentan error dividido el total de datos.
Referencia	R1
Medida	Una medida de la cantidad de datos incorrectos sobre el conjunto de datos.
Ámbito	Todas la tuplas.

Unidad de Medida	Por ciento
Tipo de medida	Evaluación, Discreta
Dimensiones relacionadas	Validez

Tabla XXI Medición de la métrica Reusabilidad

Título	**Reusabilidad**
Definición	Grado en el cual los datos son simples, relevantes, accesibles y mantenidos en el tiempo
Referencia	R5, R6
Medida	Medida de cuan relevantes, accesibles, atómicos y si pueden ser mantenidos en el tiempo.
Ámbito	Todas las tuplas.
Unidad de Medida	Variable discreta
Tipo de medida	Continua, Discreta
Dimensiones relacionadas	Validez

Tabla XXII Medición de la métrica Confianza

Título	**Confianza**
Definición	Grado en el cual los datos proceden de fuentes confiables.
Referencia	R6
Medida	Medida de cuan confiable son los datos representados en el dataset
Ámbito	Todas las tuplas.
Unidad de Medida	Variable discreta
Tipo de medida	Continua, Discreta
Dimensiones relacionadas	Validez

En el caso de la reusabilidad y confianza la variable discreta A de expresa se la siguiente manera:

$$A = \{"\text{Sí}", "No"\}$$

En este caso, es una variable binaria que puede tomar los valores de “Sí” o “No” de acuerdo al cumplimiento con las reglas del negocio que se establecen para cada una.

Reglas de negocio definidas para el caso de estudio

Tabla XXIII Reglas de negocio, definidas para el caso de estudio

Regla	Descripción
R1	249600 tuplas
R2	Tupla medida consigo misma o su contraparte en otro dataset
R3	Verificación de tipos de datos permitidos (ej: cadenas, enteros, decimal); el formato(ver sección **2.2.2 Pre-procesamiento**)
R4	Utilizar datos de referencia de terceros de fuentes que se consideran confiables.
R5	Utilizar muestras representativas de datos en el dataset
R6	Valoración subjetiva de los proveedores de los datos

3.3. Análisis de los resultados

El método para extracción, limpieza y publicación de datos de fuentes de energía renovable se procesan 19 200 filas de datos, y se genera un *dataset* con 249 600 tuplas en un formato estándar. En cada fase se aplica una o varias métricas definidas para medir la calidad de los datos que se van generando. En la Tabla XXI se muestra la relación entre las fases y las métricas aplicadas.

Tabla XXIV Relación fases-métricas

Fases	**Métricas**
Extracción	Exactitud, Confianza
Limpieza	Consistencia, Completitud, Libre-de-Error, Validez, Unicidad
Publicación	Reusabilidad

En la evaluación de los resultados se tienen en cuenta las métricas que proponen (Loshin 2011; Marev, Compatangelo y Vasconcelos 2018) para la validación de métricas de calidad de los datos. A continuación, se definen cómo se miden las dimensiones utilizadas para evaluar la calidad de los datos durante la investigación.

- **Exactitud:** Se definen tres rangos de acuerdo a dos umbrales; entre 60 y 74 por ciento, se define como *Poco Exacto*, mayor que 75 por ciento se define como *Exacto*, y *No Exacto*, en otro caso.
- **Confianza:** Se definen dos posibles valores; *De confianza* para “Sí” y *Poco confiable* para “No”
- **Consistencia:** Valores menores que 100 por ciento indican duplicidad de la información.
- **Completitud:** Valores menores que 100 por ciento indican presencia de valores nulos o blancos.
- **Libre-de-Error:** Se definen cuatro rangos de acuerdo a tres umbrales; entre 50 y 64 por ciento, se define como *Con Errores*, entre 75 y 84 por ciento, se define como *Aceptable*, mayor que 85

por ciento se define como *Libre de Error*, y *Erróneo*, en otro caso. Los valores mayores que 85 por ciento están en el rango de error donde los valores erróneos detectados pueden no representar un error en sí. Por ese motivo se clasifica como *Libre de Error*.

- **Validez:** Valores menores que 100 por ciento indican datos que no presentan la estructura definida correctamente.
- **Unicidad:** Valores menores que 100 por ciento indican que una tupla no fue almacenada más de una vez en el *dataset*.
- **Reusabilidad:** Se definen dos posibles valores; *Reusable* para "Sí" y *No reusable* para "No".

Las métricas se aplican indistintamente al principio o al final de cada fase según la medición que se desea realizar. Los cálculos y mediciones se determinan según el enfoque seguido, los resultados se muestran en la Tabla XXII.

Tabla XXV Resultados de aplicar las métricas definidas

Métrica	**Resultado**
Completitud	100%
Confianza	Sí
Consistencia	100%
Exactitud	100%
Libre-de-Error	92%
Reusabilidad	Sí
Unicidad	100%
Validez	100%

El análisis de la tabla anterior permite determinar, a partir de los resultados, que los datos generados describen las mediciones que representan el *dataset* y son de confianza, teniendo en cuenta los valores obtenidos para las métricas *Exactitud* y *Confianza*. No se poseen valores nulos o en blanco o con una estructura anómala como reflejan los valores alcanzados para las dimensiones *Completitud* y *Validez*. Existe ausencia de duplicidad de los datos y en las tuplas almacenadas, teniendo como base los resultados para las métricas *Unicidad* y *Consistencia*. El valor de la dimensión *Libre-de-Error* muestra que, aunque no se alcanza un valor de 100 por ciento, el resultado obtenido se puede clasificar como "*Libre de Error*" de acuerdo a las reglas que se definen para la evaluación de las métricas de calidad propuestas. Los datos generados son reusables teniendo en cuenta el valor resultante para la métrica *Reusabilidad.*

La interpretación de los resultados obtenido permite concluir que los datos generados poseen una alta calidad. En este contexto poseer datos con alta calidad impacta de manera efectiva sobre el proceso de toma de decisiones, haciéndolo más eficiente.

3.4. Conclusiones parciales

El uso de un enfoque de desarrollo dirigido por pruebas permitió realizar pruebas unitarias y de integración de los módulos y componentes implementados. Como resultado se obtuvo un código consistente y con baja presencia de error, permitiendo el manejo ante excepciones que puedan existir. El enfoque brindó un marco de trabajo referencial para un desarrollo ordenados durante la fase de implementación.

El caso de estudio permitió evaluar el método propuesto a partir de los resultados obtenidos mediante su ejecución en un prototipo funcional implementado. Las métricas empleadas para medir la calidad en cada fase permitieron verificar la calidad de los datos generados. El desarrollo de cada fase del método se demostró la utilidad del mismo en la extracción, limpieza y publicación de datos de fuentes de energía renovable.

CONCLUSIONES GENERALES

El estudio de las técnicas y algoritmos utilizando la revisión sistemática de la literatura evidenció la existencia de una amplia área del conocimiento en temas de extracción, limpieza y publicación de datos. Sin embargo, cada una se investiga de manera independiente pudiéndose encontrar trabajos que relacionan las dos primeras principalmente, no así métodos que integren las tres.

Las principales aproximaciones para la extracción de datos de centran en procesos de Extracción, Transformación y Carga (ETL) y en métodos que se centran solo en extracción y transformación. La limpieza de datos posee un amplio conjunto de técnicas, métodos y algoritmos que dependen de datos de entrada, datos de salida, tipo de error a detectar y manera de manejar dichos errores. La publicación de datos en la web en el contexto actual se centra en el uso de servicios web principalmente y la publicación de datos como datos enlazados utilizando los estándares de la web semántica.

En el caso de la extracción y limpieza se seleccionó el lenguaje de programación Python para las tareas de transformación y análisis de datos basándose en las experiencias de trabajos anteriores en la comunidad científica asociada a estos temas. En el caso de la extracción se utiliza una aproximación donde los datos extraídos son almacenados en una base de datos. La limpieza se centra en la detección y manejo de anomalías, siendo uno de los principales problemas que se detectan en los *dataset* durante la investigación. Con la publicación de los datos, se aplican dos enfoques: (1) orientado a servicios web y (2) orientados a consumo de datos enlazados.

El enfoque utilizado en el desarrollo del prototipo funcional correspondiente al método propuesto permitió hacer un seguimiento en cada fase de los módulos/componentes desarrollados. Se aplicaron pruebas unitarias y de integración desde las fases iniciales de la implementación del prototipo funcional.

Las métricas definidas para medir la calidad en los datos permitieron validar el método propuesto. Se analizaron las principales medidas para garantizar la calidad durante cada una de las fases del método, garantizando una fuente de datos confiable para la toma de decisiones.

RECOMENDACIONES

Aplicar técnicas de programación concurrente para lograr el análisis de varios *datasets* simultáneamente.

Definir nuevos modelos para la inclusión de nuevos *datasets* de otras energías renovables.

REFERENCIAS BIBLIOGRÁFICAS

AGUINIS, H., GOTTFREDSON, R.K. y JOO, H., 2013. Best-Practice Recommendations for Defining , Identifying , and Handling Outliers. , vol. 16, no. 2, pp. 270-301. DOI 10.1177/1094428112470848.

ALLAHYARI, M., TRIPPE, E.D. y GUTIERREZ, J.B., 2017. A Brief Survey of Text Mining : Classification , Clustering and Extraction Techniques. ,

ALVARADO, N.G., 2013. ENERGIAS RENOVABLES EN ACORDE CON EL MEDIO AMBIENTE. , vol. 200.

ANDERSEN, O., THOMSEN, C. y TORP, K., 2018. SimpleETL : ETL Processing by Simple Specifications ∗. ,

BANO, H. y AZAM, F., 2015. Innovative Windows for Duplicate Detection TT -. *International Journal of Software Engineering and Its Applications*, vol. 9, no. 1, pp. 95-104. ISSN 1738-9984.

BERTENS, R., 2015. Beauty and Brains : Detecting Anomalous Pattern Co-Occurrences. ,

CALABRESE, B., 2018. Data Cleaning. , pp. 1-4. DOI 10.1016/B978-0-12-809633-8.20458-5.

CIMIANO, P. y BIELEFELD, U., 2016. Knowledge Graph Refinement : A Survey of Approaches and Evaluation Methods. , vol. 0.

CUBAENERGÍA, 2015. Energía renovable en América Latina y el Caribe. , vol. 0.

DHRUV GAIROLA, B.S., 2015. Data Cleaning with Minimal Information Disclosure. ,

EDEKI, C., 2013. Agile unified process. *International Journal of Computer Science*, vol. 1, no. 3.

FACEBOOK OPEN GROUP, 2019. React. A JavaScript library for building user interfaces. .

GROUP, D.U.W., 2013. THE SIX PRIMARY DIMENSIONS FOR DATA QUALITY ASSESSMENT. Defining Data Quality Dimensions. ,

HAMMOND, S. y UMPHRESS, D., 2015. Test Driven Development : The State of the Practice. , no. November. DOI 10.1145/2184512.2184550.

HIGHSOFT, 2019. Interactive JavaScript charts for your webpage | Highcharts. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: https://www.highcharts.com/.

IDEA, I., 2019. IntelliJ IDEA. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: https://www.jetbrains.com/idea/features/.

ISABEL, C., DELGADO, M., MU, Y., DRC, A. y GONZ, R., 2017. *Trabajo de Diploma*. S.l.: s.n.

ISO, 2018. INTERNATIONAL STANDARD ISO / IEC / IEEE Systems and software engineering — engineering. , vol. 2018.

KNIBERG, H., 2015. *Scrum and XP from the Trenches*.

LOSHIN, D., 2011. Monitoring Data Quality Performance Using Data Quality Metrics. ,

MAREV, M.S., COMPATANGELO, E. y VASCONCELOS, W., 2018. Towards a context-dependent numerical data quality evaluation framework Technical Report. , pp. 1-12.

MYERS, G.J.., BADGETT, T. y SANDLER, C., 2012. *The Art of Software Testing*. S.l.: s.n. ISBN 9781118031964.

NACIONES UNIDAS, 2015. Agenda 2030 y los Objetivos de Desarrollo Sostenible Una oportunidad para América Latina y el Caribe. ,

NUMPY DEVELOPER, 2019. NumPy — NumPy. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: https://www.numpy.org/.

PIPINO, L.L., LEE, Y.W. y WANG, R.Y., 2002. Data Quality Assessment. , vol. 45, no. 4, pp. 211-218.

PRESSMAN, R.S., 2010. *Ingeniería del software. Un enfoque práctico*.

PYTHON SOFTWARE FUNDATION, 2019a. pandas · PyPI. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: https://pypi.org/project/pandas/.

PYTHON SOFTWARE FUNDATION, 2019b. xlrd · PyPI. [en línea]. [Consulta: 20 mayo 2019]. Disponible en: https://pypi.org/project/xlrd/.

RAMADAN, B., CHRISTEN, P. y LIANG, H., 2014. Dynamic Sorted Neighborhood Indexing for Real-Time Entity Resolution. , pp. 1-2.

SUMRA, M.B., 2018. Survey on Integration Testing of Software Components. , no. September.

SUTHERLAND, J., 2016. *Scrum: El arte de hacer el doble de trabajo en la mitad de tiempo*. S.l.: s.n.

URMA, R., FUSCO, M. y MYCROFT, A., 2015. *Java in Action*.

VILLARREAL LARRINAGA, O.;LANDETA RODRÍGUEZ, J., 2010. EL ESTUDIO DE CASOS COMO METODOLOGÍA DE INVESTIGACIÓN CIENTÍFICA EN DIRECCIÓN Y ECONOMÍA DE LA EMPRESA. UNA APLICACIÓN A LA INTERNACIONALIZACIÓN. , vol. 16, pp. 31-52.

VOLKOVS, M. y CHIANG, F., 2016. Continuous Data Cleaning. ,

WANDHEKAR, V., 2015. Proof of Duplication Detection in Data By Applying Similarity Strategies. , pp. 429-434.

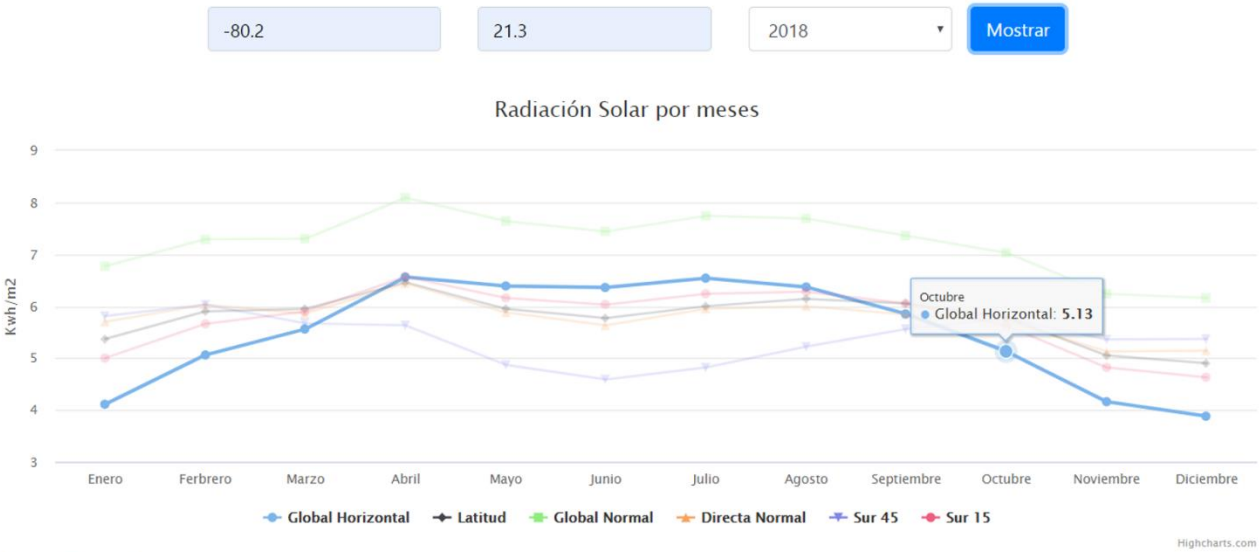
WANG, Z., HONG, J., LIU, P. y ZHANG, L., 2017. Voltage fault diagnosis and prognosis of battery systems based on entropy and Z-score for electric vehicles Voltage fault diagnosis and prognosis of battery systems based on entropy and Z -score for electric vehicles. *Applied Energy*, no. October. ISSN 0306-2619. DOI 10.1016/j.apenergy.2016.12.143.

ZHANG, Y., CHOU, P. y CHURCHILL, T., 2018. Scalable Entity Resolution Using Probabilistic Signatures on Parallel Databases. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM*

ANEXOS

Anexo 1. Interfaces del método implementado.

Anexo 2. Interfaces del método implementado.



Mes: Octubre

Anexo 3. Interfaces del método implementado.

Valores de Radiación Solar sobre Superficies con varias inclinaciones (Kwh/m2)

Nombre	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	Anual
Global Horizontal	4.11	5.06	5.56	6.56	6.39	6.36	6.54	6.37	5.85	5.13	4.16	3.88	5.49
Latitud	5.37	5.9	5.95	6.46	5.95	5.77	6	6.14	6.05	5.78	5.05	4.9	5.74
Global Normal	6.77	7.29	7.3	8.09	7.64	7.44	7.74	7.69	7.36	7.03	6.24	6.16	7.19
Directa Normal	5.7	6.03	5.86	6.44	5.88	5.63	5.95	6	5.84	5.71	5.13	5.14	5.73
Sur 45	5.81	6.03	5.67	5.63	4.87	4.59	4.82	5.22	5.56	5.76	5.36	5.37	5.34
Sur 15	5	5.66	5.9	6.57	6.16	6.03	6.24	6.28	6.05	5.62	4.82	4.63	5.72
Temperatura	26.7	27.3	28.2	29.2	30.8	30.8	31.6	31.7	31.1	30	28.9	28	29.4

Tabla XXVI Historia de usuario: Pre-procesamiento de datos

Código: ***HU-01***	Nombre del requisito: **Pre-procesamiento de datos**
Programador*:* *Tahimí González Oliva*	Iteración asignada:1
Prioridad*: alta*	Tiempo estimado: 2 semanas
Riesgo en desarrollo*: alto*	Tiempo real: 3 semanas
Descripción*: Esta tarea tiene como objetivo extraer y transformar los datos que se almacenan en hojas de cálculo que es la entrada del método. La transformación se realiza hacia un formato de tupla (ver sección* ***2.2.2 Pre-procesamiento****)*	
Prototipo de Interfaz	

Tabla XXVII Historia de usuario: Detección de anomalías

Código: ***HU-02***	Nombre del requisito: **Detección de anomalías**
Programador*:* *Tahimí González Oliva*	Iteración asignada:1
Prioridad*: alta*	Tiempo estimado: 4 semanas
Riesgo en desarrollo*: alto*	Tiempo real: 5 semanas
Descripción*: Esta tarea tiene como objetivo la detección de anomalías en el dataset generado en la etapa de pre-procesamiento. Las anomalías se detectan utilizando dos algoritmos orientados a este tipo de función (ver sección* ***2.2.3 Detección de anomalías****)*	
Prototipo de Interfaz	

Tabla XXVIII Historia de usuario: Manejo de anomalías

Código: ***HU-03***	Nombre del requisito: **Manejo de anomalías**
Programador*:* *Tahimí González Oliva*	Iteración asignada:1
Prioridad*: alta*	Tiempo estimado: 2 semanas
Riesgo en desarrollo*: alto*	Tiempo real: 1 semana
Descripción*: Esta tarea tiene como objetivo el manejo de anomalías detectadas en la etapa de pre-procesamiento (ver sección* ***2.2.4 Manejo de anomalías****).*	
Prototipo de Interfaz	

Tabla XXIX Historia de usuario: Exportar datos

*Código: **HU-04***	Nombre del requisito: **Exportar datos**
Programador*:* *Tahimí González Oliva*	Iteración asignada:1
Prioridad*: alta*	Tiempo estimado: 1 semanas
Riesgo en desarrollo*: alto*	Tiempo real: 1 semana
Descripción*: Esta tarea tiene como objetivo exportar los datos generados a una base de datos relacional (ver sección **2.2.5 Exportar datos**).*	
Prototipo de Interfaz	

Tabla XXX Historia de usuario: Publicar datos

*Código: **HU-05***	Nombre del requisito: **Publicar datos**
Programador*:* *Tahimí González Oliva*	Iteración asignada:1
Prioridad*: alta*	Tiempo estimado: 2 semanas
Riesgo en desarrollo*: alto*	Tiempo real: 2 semana
Descripción*: Esta tarea tiene como objetivo publicar los datos almacenados en la base de datos utilizando servicios web, además de proveer de un SPARQL Endpoint(ver sección **2.2.6 Publicación datos**).*	
Prototipo de Interfaz	