UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Tecnura

https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/issue/view/1108

DOI: https://doi.org/10.14483/22487638.17246

INVESTIGACIÓN

Metodología de desarrollo de técnicas de agrupamiento de datos usando aprendizaje automático

Development Methodology of Techniques for Data Clustering Using Machine Learning

Ghiordy Ferney Contreras Contreras [1], Byron Medina Delgado [2], Brayan René Acevedo Jaimes [3], Dinael Guevara Ibarra [4]

Fecha de Recepción: 09 de septiembre de 2020 **Fecha de Aceptación**: 07 de febrero de 2022

Cómo citar: Contreras-Contreras., G.F. Medina-Delgado., B. Acevedo-Jaimes., B.R y Guevara-Ibarra., D. (2022) Metodología de desarrollo de técnicas de agrupamiento de datos usando aprendizaje automático. *Tecnura*, **26**(72), 42-58. https://doi.org/10.14483/22487638.17246

Resumen

Contexto: Hoy en día, el uso de grandes cantidades de datos adquiridos desde diversos dispositivos y equipos electrónicos, ópticos u otra tecnología de medición, generan un problema de análisis de datos en el momento de extraer la información de interés desde las muestras adquiridas. En ellos, agrupar correctamente los datos es necesario para obtener información relevante y precisa para evidenciar el fenómeno físico que se desea abordar.

Metodología: El trabajo presenta la evolución de una metodología de cinco etapas para el desarrollo de una técnica de agrupamiento de datos, a través de técnicas de aprendizaje automático e inteligencia artificial. Esta se compone de cinco fases denominadas análisis, diseño, desarrollo, evaluación y distribución, con estándares de código abierto y fundamentadas en los lenguajes unificados para la interpretación del *software* en ingeniería.

Resultados: La validación de la metodología se ha desarrollado mediante la creación de dos métodos de análisis de datos, con un tiempo de ejecución promedio de 20 semanas, obteniendo valores de precisión 40 % y 29 % superiores con los algoritmos clásicos de agrupamiento de datos de *k-means* y *fuzzy c-means*. Adicionalmente, se encuentra una metodología de experimentación masiva sobre pruebas unitarias automatizadas, las cuales lograron agrupar, etiquetar y validar 3,6 millones de muestras, acumulado un total de 100 ejecuciones de grupos de 900 muestras, en aproximadamente 2 horas.

[1]Ingeniero electrónico. Estudiante de Maestría en Ciencias en el Centro de Investigación de Estudios Avanzados del Instituto Politécnico Nacional, Guadalajara, México.
Email: ghiordyferneycc@ufps.edu.co
[2]Ingeniero electrónico, magíster en Ingeniería Electrónica, doctor en Ciencias. Profesor de la Universidad Francisco de Paula Santander. Cúcuta, Colombia.
Email: byronmedina@ufps.edu.co
[3]Ingeniero electrónico, magíster en Ingeniería Eléctrica. Investigador Computational Intelligence Laboratory, LITC, Belo Horizonte, Brasil.
Email: payo.rene@ufmg.br
[4]Ingeniero electricista, especialista en Teleinformática, magíster en Ingeniería Electrónica, doctor en Ingeniería. Profesor de la Universidad Francisco de Paula Santander, Cúcuta, Colombia.
Email: dinaelgi@ufps.edu.co

Conclusiones: Con los resultados de la investigación se ha determinado que la metodología pretende orientar el desarrollo sistemático de técnicas de agrupamiento de datos, en problemas específicos para bases integradas por muestras con atributos cuantitativos, como los casos de parámetros de canal en un sistema de comunicaciones o la segmentación de imágenes usando los valoras RGB de los pixeles; incluso, cuando se desarrolla *software* y *hardware*, la ejecución será más versátil que en casos con aplicaciones teóricas.

Palabras clave: análisis de datos, automatización, algoritmo, *software* de código abierto.

Financiamiento: Universidad Francisco de Paula Santander y Univeridade Federal de Minas Gerais.

Abstract

Context: Today, the usage of large amounts of data acquired from various electronic, optical, or other measurement devices and equipment brings the problem of data analysis at the time of extracting the aimed information from the acquired samples. Where to correctly group the data is necessary to obtain relevant and accurate information to evidence the physical phenomenon that you want to address.
Methodology: The work presents the development and evolution of a five-stage methodology for the development of a data grouping technique, using machine learning techniques and artificial intelligence. It consists of five phases called analysis, design, development, evaluation, and distribution, using open-source standards, and based on unified languages for the interpretation of software in engineering.
Results: The validation of the methodology was developed through the creation of two data analysis methods, with an average execution time of 20 weeks, obtaining precision values 40 % and 29 % higher with the classic data grouping algorithms of k-means and fuzzy cmeans. Additionally, there is a massive experimentation methodology on automated unit tests, which managed to group, label, and validate 3.6 million samples accumulated in the total of 100 group runs of 900 samples in approximately 2 hours.
Conclusions: Finally, with the results of the research was determined that the methodology intends to guide the systematic development in specific problems in quantitative databases, such as the channel parameters in a communication system or the segmentation of images using the RGB values of the pixels. Even when software is developed both hardware, the execution will be more versatile than in cases with theoretical applications.

Keywords: data analysis, automation, algorithm, open-source software.

Financing: Universidad Francisco de Paula Santander and Universidade Federal de Minas Gerais.

Tabla de Contenidos

Introducción

El área de reconocimiento de patrones poblacionales en bases de datos aborda los problemas de vanguardia en la industria y la automatización de procesos (Babic *et al.*, 2008); se presenta como una herramienta de la inteligencia artificial (IA), acompañando el aprendizaje automático y la visión por computadora. Estas áreas han logrado avances en materia de diagnóstico, predicción e identificación de las características cuantitativas o cualitativas con las cuales automatizar procesos (Jain *et al.*, 1999). La IA, dentro de su rango de aplicaciones (Hernández *et al.*, 2021, Luque *et al.*, 2020, Ramírez-Escobar *et al.*, 2021, Giral *et al.*, 2021, Sánchez-Quintero *et al.*, 2021), permite tomar decisiones objetivamente con aprendizaje automático, basadas en las variables cuantitativas de un proceso industrial, y en la experiencia reciente de las entradas y salidas del proceso (Akyol, 2020).

En Colombia, la Superintendencia de Industria y Comercio está implementando técnicas populares de *machine learning* (ML) para la clasificación de sus clientes, manipulación de datos y movimientos de la economía (Moreno, 2009). Además, aplica la IA a problemas ambientales y de optimización del recurso hídrico en la ciudad de Bogotá (Solano Meza *et al.*, 2019). Otros trabajos llegan a usar las herramientas tecnológicas emergentes en problemas de simulación abordados en la formación de pregrado de electrónica analógica (Ramírez-Carvajal *et al.*, 2019) y sistemas de radiocomunicaciones terrestres (Báez Perez y Soto-Vergel, 2019).

En el trabajo presentado por Gasca *et al.*, 2014 se propone una metodología para el desarrollo de aplicaciones móviles con salidas rápidas al mercado, a partir de metodologías ágiles. De igual

forma, (Molina *et al.*, 2010) presentan la relación entre el analista de requerimientos y los interesados comercialmente en el proyecto, usando el lenguaje gráfico *business process management notation* (BPMN), definiendo las necesidades inmediatas del cliente o consumidor, soportándose sus proposiciones lógicas con el estándar *unified modelling language* (UML) y evitando la inconsistencia del modelado entidad/cliente (Lucas *et al.*, 2009).

Actualmente, se cuenta con técnicas de aprendizaje automático para aplicaciones de ingeniería, como el pronóstico de radiación solar (Diagne *et al.*, 2013), en ciencias exactas, con la caracterización de materiales de manera automatizada (Ong *et al.*, 2019), donde el objetivo consiste en extraer información de similitud cuantitativa a través de la matriz de distancias para agrupar las muestras, sin información *a priori* de la similitud con la cual determinar los resultados que se deberían obtener en un proceso entrada/salida de clasificación (Jaimes *et al.*, 2017).

Este trabajo presenta la propuesta de una metodología para el desarrollo de técnicas de agrupamiento de datos, reconociendo los aspectos de desarrollo técnico de las metodologías agiles y rápida salida al mercado (Amaya Balaguera, 2015), con la cual se quiere evidenciar la importancia de definir los factores medibles y recursivos en cada fase de la investigación de agrupamiento de datos (Gargiulo *et al.*, 2018), reduciendo el tiempo en la toma de decisiones y elaborando la revisión automática.

METODOLOGÍA

El trabajo se fundamenta en la experiencia de las investigaciones previas en reconocimiento de patrones, donde la evaluación de la metodología ha sido medida a través de tiempos de ejecución consumidos para entregar un resultado o producto. Al obtener estos resultados, se aplicaron índices de error y precisión para validar la calidad de estos con la metodología enfocada a la salida al mercado del producto, comparando los resultados con otras enfoques y trabajos publicados en la literatura. Para esto, se han definido cinco etapas de ejecución presentadas en la figura **1** de manera metódica: análisis, diseño, desarrollo, evaluación y distribución. Debido a que cada aplicación debe enmarcarse en el desarrollo ágil y eficaz (Gasca *et al.*, 2014), reduciendo los tiempos de salida al mercado tanto como sea posible.

De esta manera, cada etapa describe actividades enmarcadas en procesos metodológicos, ordenados y con un objetivo claro en cada una de ellas, es decir, con metas claras que el equipo de trabajo pueda identificar las actividades faltantes para completar la etapa.

Análisis

Variables

En el tratamiento de datos inicialmente se describe el conjunto de datos con el objeto de obtener las variables (o atributos) con las que cuenta, además del número de muestras allí contenidas. Para

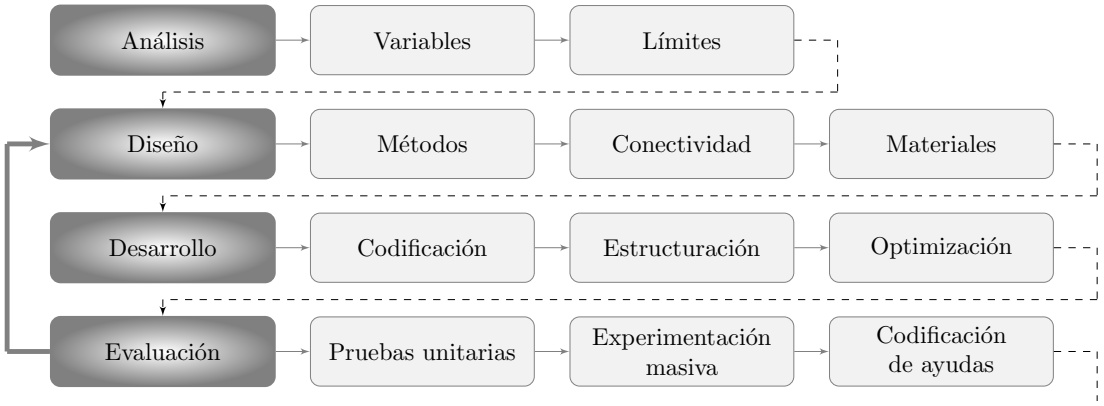

Figura 1. Estructura general de la metodología

Fuente: elaboración propia.

esto, la estadística descriptiva ha dispuesto las herramientas básicas para esta tarea, como la media aritmética, el valor mínimo, el valor máximo, los cuartiles y la desviación estándar, describiendo los datos en general. Posteriormente, se profundiza en el análisis estadístico con los histogramas, en los cuales se encuentran las distribuciones o forma del conjunto de datos.

Límites

En consecuencia, bajo mediciones reales hay consideraciones técnicas afines con la estrecha relación de las medidas con la teoría física, de la cual se extraen los límites o criterios que deben seguir las muestras.

Teniendo en cuenta los límites y variables, se proponen los siguientes pasos para analizar el conjunto de datos abordado:

- Identificar el formato digital del archivo, la cantidad de muestras y atribuciones, además de depurar las muestras, es decir, aquellos datos que constituyen errores de formato o lectura deben ser retirados del conjunto de datos.
- Seguidamente se establece la relación teórica, que algunos autores han descrito previamente sobre las variables, en el caso de contar con esta. En caso contrario, solo se usan los umbrales máximos y mínimos de medición definidos por el equipo de medición para la adquisición de los datos.
- Finalmente, al contar con las variables, la relación existente entre ellas, así como también, los umbrales teóricos y prácticos que presentan los datos, se procede con la normalización de los datos bajo los criterios previamente extraídos desde el aspecto real y el desarrollo teórico.

Diseño

Tomando una arquitectura modular para el diseño del sistema de agrupamientos de datos, es necesario ordenar los métodos, así como las funcionalidades de cada uno y la conectividad o flujo continuo de la información entre ellos. Todo esto depende de la técnica de agrupamiento, debido a que en la literatura se presentan varios enfoques desde el particional como en el trabajo de Zhu y Ma, 2018; el jerárquico, en el trabajo de (Gilbert *et al.*, 2020), y el no supervisado, en los trabajos de (Kwon *et al.*, 2018), (Mwangi *et al.*, 2014) y (Tafsast *et al.*, 2017).

Métodos

A medida que aumenta la complejidad para agrupar datos debido a su naturaleza, puede existir un mayor número de etapas que componen la metodología. La mayoría de las veces, estas etapas son destinadas al pre- o posprocesamiento de los datos. De forma similar, en esta metodología se proponen las siguientes etapas: definir un módulo de lectura, preprocesamiento y valoración del conjunto de datos, estructurar un módulo para llevar el enfoque de agrupamiento que se desarrolla, evaluar, almacenar e interpretar los resultados usando medidas extraídas de la matriz de distancias.

Conectividad

Con los métodos generales del enfoque metodológico, adicionalmente, se debe conectar en secuencia idónea para evitar la pérdida de información o manipulación de esta, al tener solo tres métodos generales, cada uno de estos presenta más módulos, incluso para cualquier enfoque de agrupamiento usado.

Materiales

Por último, se enlistan los materiales de *software* y *hardware* disponibles para el desarrollo, teniendo en cuenta que, al elevar los costos de desarrollo, afectará directamente los tiempos para culminar el producto. Para tal caso se recomienda:

- Realizar el inventario de las herramientas de *hardware*, como computadores con su configuración completa de periféricos de entrada (ratón, teclados) y periféricos de salida (monitores o proyectores, incluso adaptadores USB a tarjetas de desarrollo), tarjetas de video y sistemas basados en microcontroladores.
- Realizar inventario de las herramientas de *software*, generalmente para el caso de agrupamiento de datos las opciones como usar lenguajes R o Python, permiten que la mayor parte de los procedimientos en agrupamiento de datos sean abordados con las librerías de *scikit-learn, SciPy, Pandas, Seaborn, NumPy*, entre otros.

Desarrollo

Inicialmente, cuando se desarrollan aplicaciones de *software*, se integran varias personas en el equipo de trabajo; ellas deben contribuir ordenadamente y con criterios grupales, bajo los cuales usar herramientas como *GitHub* o *GitLab*, permiten un flujo de desarrollo secuencial y escalable, cada cambio queda registrado históricamente en el repositorio web de las herramientas, además de la manipulación de varias configuraciones de máquina o sistemas operativos diferentes, para los roles de cada miembro del equipo. Por esta razón, antes de codificar se debe estructurar la ubicación del enfoque con el objeto de desarrollar eficazmente sobre un trayecto claro, donde la posterior fase de codificación consiste en definir un módulo principal para cada archivo ejecutable con una tarea específica. Para esto, se usa archivos ejecutables con funciones específicas (*single responsability principle*, SRP), pero añadiendo dentro de cada una los umbrales de ejecución requeridos. Además, se menciona la importancia de abordar un paradigma de programación específico, al nivel de la abstracción del enfoque, teniéndose generalmente los siguientes paradigmas para esta tarea, entre los cuales se enmarca la mayor cantidad de aplicaciones de *software*:

- *Paradigma de programación estructurada:* abordado en aplicaciones sencillas, donde el flujo de la información es lineal y adicionalmente elaborado bajo módulos individuales de ejecución para cada una de las tareas desarrolladas.

- *Paradigma de programación orientada a objetos:* para aplicaciones donde los datos presentan variedad de atribuciones operativas, clasificaciones entre grupos, y relaciones indirectas de cada una de estas; generalmente usado para aplicaciones cotidianas, donde los datos son ambiguos, y se requiere un mayor análisis o caracterización para agrupar las muestras a un conjunto.

Otros paradigmas de programación suelen ser combinaciones de estos o arreglos con otras metodologías de desarrollo de *software*; sin embargo, para el área de agrupamiento de datos, con la programación estructurada o procedimental es suficiente para la estructuración del enfoque, aunque no todos los conjuntos de datos se ajustan a estos.

Por último, después de aplicar los métodos de desarrollo y estructuración, se debe optimizar el enfoque de múltiples o diversas formas, para lo cual se ha planteado en la figura **2** un modelo de ejecución de cada una de las propuestas presentadas por el equipo de trabajo, donde inicialmente se elabora el concepto de la idea; dentro de una discusión (o debate) se verifica la factibilidad de esta idea con respecto a la propuesta de valor presente en el diseño, desde los cuales se deben tener presente los requerimientos de usuario o abordarla donde posteriormente se ajusta la propuesta. Una vez que la propuesta cumpla con los criterios de factibilidad, se codifica la parte que optimiza el enfoque, con el objeto de adquirir los requerimientos y debatirlos nuevamente en el comité de valoración, verificando el producto sobre estas fases hasta que cumpla los requerimientos y posteriormente distribuir el producto o desarrollo.

Figura 2. Método de refinamiento estándar bajo criterios de funcionalidad técnica o accesible

Fuente: elaboración propia.

Evaluación

Esta sección se centra en evaluar el desempeño del enfoque mediante datos sintéticos y reales, donde cada uno presenta un comportamiento diferente, dependiendo de las situaciones teóricas planteadas para cada uno. Por esto, inicialmente se crean las rutinas de estructuración de datos sintéticos, con información preliminar suficiente para evaluar la calidad del agrupamiento, es decir, se definen los rótulos de los grupos a los cuales deben ser atribuidas cada una de las muestras. Sin embargo, la ejecución de pruebas unitarias consiste en desarrollar procesos de valoración o evaluación del código de manera automatizada, esto quiere decir que se requiere de rutinas cíclicas iterando pruebas unitarias sobre un tipo de entradas con algunos parámetros definidos previamente y bajo los cuales se establecen los cambios con el objeto de mejorar los resultados o en el caso del agrupamiento, la clasificación de las muestras (Villa Betancur y Giraldo Plaza, 2012). Por otro lado, cuando se trabaja con datos, la importancia o relevancia de los agrupamientos va con la cantidad de datos usados para la experimentación, lo que requiere abordar la experimentación masiva a través de pruebas unitarias automatizadas, teniendo en cuenta los siguientes elementos para el reporte de resultados: precisión en la predicción obtenida, complejidad computacional, especialmente la complejidad algorítmica temporal empírica, pues la espacial no pierde relevancia, pero esta va estrechamente relacionada a los requisitos mínimos de almacenamiento, ejecución simultánea con otra metodología o algoritmo propuesto en la literatura, teniendo en cuenta que sea una técnica ampliamente usada, pero con algunas limitaciones que fueron obviadas en su desarrollo.

Teniendo estos tres componentes, la validación de los resultados es enfocada a un marco comparativo, con las condiciones iniciales bajo las cuales se desarrolla, además de registrar cada experimento, para los cuales se recomienda evitar guardar variables normalizadas y enfocarse en los resultados de la valoración. Consecuentemente, se presenta el modo de ejecución para datos reales, estos son más complicados, pues no se tiene información preliminar acerca del grupo al cual debería ser atribuido (Jaimes *et al.*, 2017), además de que iterar sobre los mismos de manera masiva no tendría resultados diferentes, para lo cual se plantea usar funciones de extracción de atributos de interés con información que permita el agrupamiento, estás funciones verifican estadísticamente que se cuenta con dos o más distribuciones de datos. Consecuentemente, se conectan las funciones a modos de ex-

perimentación unitaria y automatizada, con los cuales se evalúa el conjunto de datos, que consiste en el desarrollo de pruebas unitarias, las cuales para datos reales debe ser abordado hasta conseguir resultados estables, teniendo en cuenta las atribuciones del conjunto de datos que representan peso en la formación de grupos dentro de ellos, con el objeto de iterar, posteriormente, de manera automatizada en ejecuciones masivas y registrando los dos últimos criterios usados para datos sintéticos.

Finalmente, después de elaborar las rutinas de ejecución masiva, se han encontrado varios interrogantes al abordar la metodología, donde un usuario del enfoque se preguntaría "¿Cómo evaluar un conjunto de datos?, ¿cuáles son las funciones principales?, ¿qué se debe realizar para ingresar mis datos?". En este sentido, se elaboran ayudas como comentarios en el código, adicionando el comando *help* en el cual encontrar las instrucciones generales de ayuda, como guía para empezar a usar ágilmente el enfoque, además de reducir los tiempos para comprender el enfoque, además de la elaboración de comandos básicos y la descripción de lo que cada uno hace o cumple dentro del enfoque.

Entrega

En todo desarrollo tecnológico actual debe incluirse instrucciones a medida que completan los módulos que lo complementan (Molina *et al.*, 2010), en el caso de los códigos, el uso de comentarios *docstrings*, para Python se dispone de un estilo de comentario como documentación del código, donde se encuentra información descriptiva de las funciones, entradas y salidas para cada uno de los métodos, además de la adicción de comentarios sugiriendo cambios comunes del código. Lo primero que se le entrega a un usuario con el producto es el manual de usuario para su ejecución inmediata, donde se encuentra la descripción general o funcionalidad base, requerimientos de máquina o prerrequisitos para usarlo, un modelo de configuración rápida, las herramientas con las que cuenta y la licencia bajo la cual la persona usa el producto. Incluso cuando el enfoque es de código abierto, hay gran variedad de licencias para regular el uso de este, en el comercio, distribución y modificación, además de la garantía, el reconocimiento y el soporte por parte de los autores (o desarrolladores), y existen las posibilidades de licencias desde la Apache hasta la ampliamente usada MIT.

Después de completar la información del manual o guía de usuario, encontramos que los códigos requieren visualizar su documentación en una plataforma de acceso público, incluso dentro del *software* añadir demasiada documentación elevara indeseablemente la complejidad espacial del enfoque y reducirá el rango de máquinas que usarían el enfoque. Herramientas como la plataforma de *GitHub* o *GitLab* permiten ampliar la información acerca del desarrollo, incluso elaborando vídeos conectados a YouTube con el tutorial de inicio rápido del enfoque, innumerables referencias, imágenes y otros trabajos relacionados. Por otro lado, las plataformas mencionadas previamente parecen ser desconocidas por el público en general, pues la mayoría usa únicamente navegadores de internet para encontrar soporte acerca de aplicaciones de *software*, donde una solución ágil en el desarrollo de una plataforma web para visualizar el código sería Sphinx, donde en el caso de Python, se ajusta a la

sintaxis de los *docstrings* y comentarios para crear la documentación, con una interfaz gráfica amigable al usuario e información precisa. Finalmente, para elegir el tipo de licencia se han propuesto tres, desde la licencia MIT ampliamente citada para el reconocimiento del autor y su trabajo dentro de un proyecto aún más grande, siguiendo con la familia de licencias GNU que permiten acceso público y variando en términos de uso comercial, según sea el caso, y llegando la licencia Apache, de la cual generalmente se exalta el producto y su aplicabilidad, reduciendo las limitaciones comerciales.

Resultados

Bajo la metodología presentada se han desarrollado dos proyectos: el primero es *cluster-CV2*, el cual se estructura como un abordaje de visión computacional para identificación espacial de agrupamientos de datos, enfocado en datos con problemas de superposición de muestras provenientes de diferentes grupos; el segundo es el desarrollo de un sistema de adquisición de datos basado en Arduino y modelamiento con redes neuronales para análisis de datos, con el cual, al aplicar el *cluster-CV2* sobre las mediciones obtenidas de este dispositivo, se obtienen tres grupos sin información preliminar para la clasificación. Las muestras son medidas de un sistema térmico para incubación de aves domésticas, con las cuales se ha creado un modelo de redes neuronales con la información de los grupos, al cual cada una de las muestras nuevas debe ser atribuido. El desarrollo de los dos enfoques se presenta desde metodología usada para obtener desarrollos eficaces y tiempo de ejecución.

Agrupamiento de datos no supervisado

En el enfoque de agrupamientos de datos (Contreras Contreras, Dulcé-Moreno *et al.*, 2019) al desarrollar la fase de análisis, se encontraron los requerimientos funcionales bajo los cuales se propuso que el *cluster-CV2* debe tener un enfoque no supervisado, es decir, el usuario no conoce el número de *k*-grupos a los cuales atribuir las muestras. La identificación se realiza sobre la matriz de distancias euclidianas de los datos para después pasarla a través de operadores morfológicos de visión computacional. Las muestras con superposición entre ellas deben ser corregidas usando la propagación y dispersión de estas en el espacio de proyección lineal. Con estos problemas definidos, se procedió a desarrollar el diseño, el cual consiste en once funciones con procedimientos matemáticos para la extracción de un matriz de distancias, proyectada sobre una imagen, identificando los *k*-grupos contenidos usando operadores morfológicos sobre los píxeles de la imagen generada de la matriz de distancias, y finalmente corregir la superposición a través del análisis de componentes principales sobre cada grupo detectado. Siendo un diseño secuencial, se implementó la metodología de pruebas unitarias durante el desarrollo, notar que a medida que se pasaba por las fases del desarrollo se iban usando las fases de evaluación, y verificando las pautas del diseño planteado en los algoritmos. En la evaluación del producto, se evidenció la necesidad de usar datos en una ejecución masiva, pues la ejecución unitaria no era lo suficientemente factible para guardar registros del desempeño, bajo lo

cual se desarrolló el sistema automatizado de la figura 3, cuyos parámetros de entrada son el número de grupos sintéticos con los que se desea realizar la prueba y el nivel de ruido presente en sus grupos, de la misma forma se obtenía un reporte con los resultados de la identificación, corrección de la superposición y validación con métricas de precisión y exactitud.

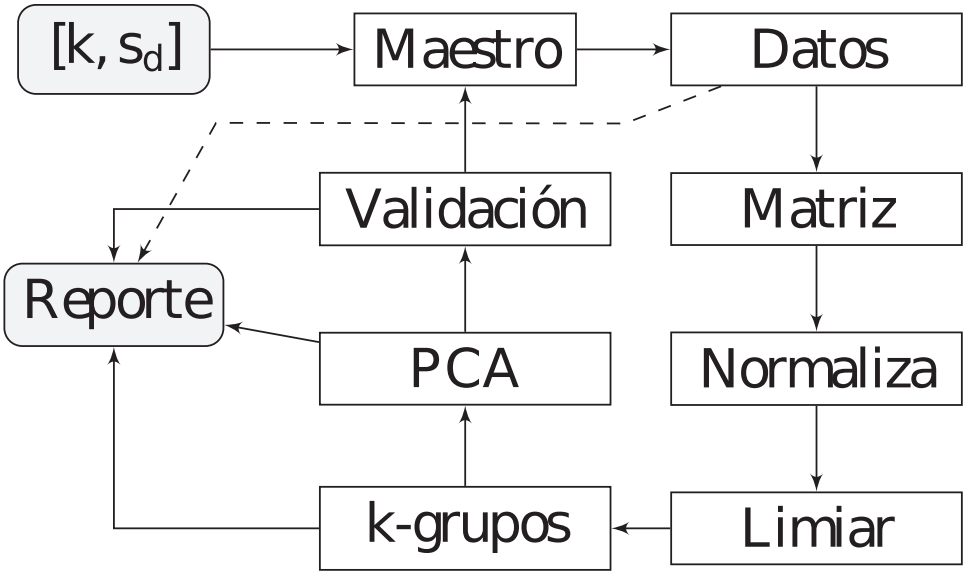

Figura 3. Método de refinamiento estándar bajo criterios de funcionalidad técnica o accesible

Fuente: elaboración propia.

Una visión acerca del sistema automatizado de experimentación es esquematizado en la figura 3 con *software* de agrupamiento, con jerarquía modular, iniciando el flujo de información con el módulo en k, s_d, obligatorio para los algoritmos de ejecución, el cual contiene los directorios locales para el acceso a bases de datos y los archivos ejecutables (bloques: *Matriz, Normaliza, Limiar, k-grupos, PCA, Validación*), cuyo modulo gestiona los resultados usando el bloque *Maestro* guardando la información de las métricas de validación, la corrección de la superposición y la identificación de los grupos en el bloque *Reporte* como archivos separados por coma (CSV).

Por consiguiente, la cuarta fase, evaluación, utilizó para este enfoque dos métricas de validación de resultados de agrupamiento como silhouette y precisión (Halkidi *et al.*, 2020), promediando los resultados obtenidos en la tabla 1, variando la desviación estándar atribuida para cada grupo de datos sintéticos construidos con normas gaussianas. Con respecto a la precisión, aplicando la metodología al *cluster-CV2* alcanza resultados superiores con respecto a los enfoques *k-means* y *c-means*, sin embargo, la métrica *silhouette* arroja resultados más altos mediante las otras dos técnicas.

Bajo estos datos se ha estructurado un enfoque de agrupamiento de datos con resultados competitivos a las técnicas clásicas de la literatura; sin embargo, para el despliegue exitoso, en la fase 5 de la metodología, se organizaron los archivos dentro de un repositorio en línea de GitHub, cuyo

Tabla 1. Comparación de resultados del *cluster-CV2* con el trabajo

Nivel de *Cluster-CV2*			*k-means*		*c-means*	
ruido	***Silhouette***	**Precisión**	***Silhouette***	**Precisión**	***Silhouette***	**Precisión**
0,1	0,6904	0,9933	0,5912	0,5541	0,8446	1,0000
0,2	0,5705	0,9933	0,6894	0,9867	0,6893	0,9845
0,3	0,4477	0,9914	0,5537	0,4498	0,5518	0,4499
0,4	0,3091	0,9662	0,4923	0,3734	0,4909	0,3722

Fuente: Tomado de Contreras Contreras, Medina Delgado *et al.*, 2019 y Jaimes *et al.*, 2017.

repositorio identifica tres secciones: la *a* describe rutinas de ejecución para datos sintéticos; la *b* contiene los archivos con datos reales, además de los análisis dentro de archivos Jupyter Notebook, y la 3 contiene la ejecución de pruebas unitarias tanto para experimentos individuales como para ejecución masiva, recopilando todas las guías dentro del manual de usuario con la cual se va elaborando la documentación dentro del mismo código, y resumiendo todos los aspectos generales del enfoque dentro del archivo README.md de GitHub, incluyendo la licencia e información general acerca de los autores, desarrolladores y colaboradores.

Sistema de adquisición de datos

La metodología fue implementada en el desarrollo de un sistema de adquisición de datos para el análisis usando redes neuronales artificiales como operadores predictivos para cambio de estado en un sistema termodinámico (Contreras Contreras, Medina Delgado *et al.*, 2019). La definición de requerimientos abordó solo dos elementos cruciales, desde el comportamiento teórico y los requerimientos de usuario del sistema: el registro no supervisado de las variables de temperatura y humedad relativa para los puntos de fuente de calor y carga del sistema termodinámico, variando la referencia de control de manera que permita predecir la acción que evitará la variación desde el estado de estabilidad.

Teniendo claro el enfoque, la figura **4** presenta el diseño preliminar del sistema planta-controlador, el cual consta de una interacción directa en planta, pues los datos se registran desde la fuente, el control y la carga, además de usarlos en el control del suministro energético de la fuente hacia la carga.

En el desarrollo se elaboró una serie de algoritmos para cumplir con los requerimientos, dentro de los cuales se van depurando sus declaraciones usando pruebas unitarias, llegando a la experimentación masiva y documentación del trabajo. Sin embargo, aunque este es un enfoque con parte *hardware* y parte *software*, se evidencia en la figura **5** que las primeras dos fases requirieron cuatro semanas con

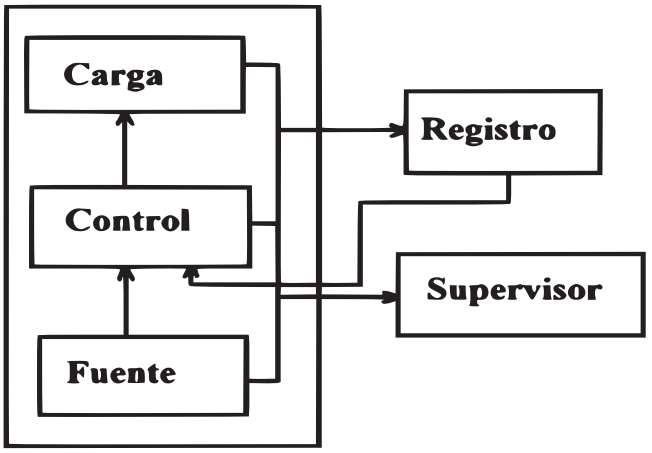

Figura 4. Diseño de la interacción del sistema termodinámico con el enfoque de adquisición de datos

Fuente: elaboración propia.

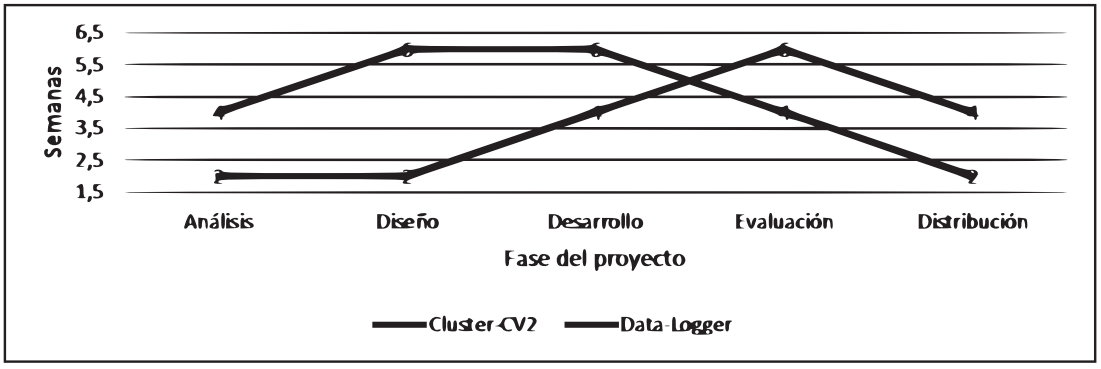

Figura 5. Comparativa de tiempos en el desarrollo de los dos productos

Fuente: elaboración propia.

respecto a las diez del *cluster-CV2*, debido a dos factores esenciales: la necesidad inmediata y relación directa de los trabajos dentro de las áreas de inteligencia artificial y aprendizaje no supervisado, y de la experiencia obtenida previamente aplicando la metodología en aplicaciones para aprendizaje no supervisado.

Por otro lado, al incluir un sistema de adquisición de datos para la experimentación real, al contener dos fases de desarrollo (*software* y *hardware*), se requieren mayores esfuerzos en la evaluación de resultados, pues si bien la metodología de pruebas unitarias lograría ser aplicada efectivamente para experimentos en implementación física, no logra la misma versatilidad en la experimentación masiva con respecto al desarrollo estrictamente teórico.

Conclusiones

Al aplicar la metodología sobre el desarrollo de enfoques de análisis y adquisición de datos se obtuvo un tiempo de 22 semanas para el trabajo teórico, en el cual no se requiere la adquisición de materiales de *hardware*, para el caso de *software* y *hardware*, se requirió de 18 semanas. En este tipo de enfoques para la reducción de tiempo de desarrollo se ha propuesto la hipótesis de una definición claramente delimitada de los alcances, reduciendo el rango de actividades para el usuario y desarrollador. Por otro lado, el sistema *cluster-CV2* ha alcanzado resultados comparables aplicando las métricas de la silhouette y precisión, donde sus variaciones con respecto a las técnicas *k-means* y *c-means* dan la perspectiva de obtener información conectada en lugar de usar el enfoque particional, incrementado la métrica de precisión a 0,6571 cuando se varía el nivel de ruido a 0,4 veces la desviación estándar en cada *k*-grupo. En general, la propuesta metodológica se fundamenta en el análisis de requerimientos funcionales, técnicos y de accesibilidad al usuario, delimitación del diseño a través de una ejecución secuencial del proyecto, con retroalimentación en fase de evaluación hacia el diseño, donde la automatización de pruebas masivas fue orientada como método de detección de errores o funcionalidades incompletas dentro de las sentencias del enfoque.

Financiamiento

Este artículo es un producto del proyecto de investigación "Cluster-CV2: un abordaje de visión computacional para identificación espacial de agrupamientos de datos", financiado por la Universidad Francisco de Paula Santander, con contrapartida de la Universidade Federal de Minas Gerais.

Agradecimientos

Los autores del trabajo expresan su agradecimiento al Laboratorio de Inteligencia y Tecnología Computacional de la Universidad Federal de Minas Gerais, el cual permitió el acceso remoto a un computador especializado para desarrollo de *software*, además del Grupo de Investigación y Desarrollo en Electrónica y Telecomunicaciones de la Universidad Francisco de Paula Santander, el cual dispuso los espacios para el desarrollo continuo del proyecto.

Referencias

[Akyol, 2020] Akyol, K. (2020). Comparing of deep neural networks and extreme learning machines based on growing and pruning approach. *Expert Systems with Applications*, 140, 112875. https://doi.org/10.1016/j.eswa.2019.112875 ↑Ver página 44

[Amaya Balaguera, 2015] Amaya Balaguera, Y. D. (2015). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. *Revista de Tecnología, 12*(2). https://doi.org/10.18270/rt.v12i2.1291 ↑Ver página 45

[Babic *et al.*, 2008] Babic, B., Nesic, N. y Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. *Computers in Industry, 59*(4), 321-337. https://doi.org/10.1016/j.compind.2007.09.001 ↑Ver página 44

[Báez Perez y Soto-Vergel, 2019] Báez Perez, A. A. y Soto-Vergel, Á. J. (2019). Enseñanza de sistemas de radiocomunicaciones terrestres con línea de vista mediante *software* educativo. *Revista Educación en Ingeniería, 14*(28), 78-87. ↑Ver página 44

[Contreras Contreras, Dulcé-Moreno *et al.*, 2019] Contreras Contreras, G. F., Dulcé-Moreno, H. J. y Melo, R. A. (2019). Arduino data-logger and artificial neural network to data analysis. *Journal of Physics: Conference Series*, 1386, 12070. https://doi.org/10.1088/1742-6596/1386/1/012070 ↑Ver página 51

[Contreras Contreras, Medina Delgado *et al.*, 2019] Contreras Contreras, G. F., Medina Delgado, B., Ibarra, D. G., Leite De Castro, C. y Acevedo Jaimes, B. R. (2019, April 1). Cluster CV2: A computer vision approach to spatial identification of data clusters. En *2019 22nd Symposium on Image, Signal Processing and Artificial Vision, STSIVA 2019 - Conference Proceedings* (pp. 1-5). https://doi.org/10.1109/STSIVA.2019.8730239 ↑Ver página 53

[Diagne *et al.*, 2013] Diagne, M., David, M., Lauret, P., Boland, J. y Schmutz, N. (2013). Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, 27, 65-76. https://doi.org/10.1016/j.rser.2013.06.042 ↑Ver página 45

[Gargiulo *et al.*, 2018] Gargiulo, F., Silvestri, S. y Ciampi, M. (2018). A clustering based methodology to support the translation of medical specifications to software models. *Applied Soft Computing Journal*, 71, 199-212. https://doi.org/10.1016/j.asoc.2018.03.057 ↑Ver página 45

[Gasca *et al.*, 2014] Gasca Mantilla, M. C., Camargo Ariza, L. L. y Medina Delgado, B. (2014). Metodología para el desarrollo de aplicaciones móviles. *Tecnura, 18*(40), 20-35. ↑Ver página 44, 45

[Gilbert *et al.*, 2020] Gilbert, N., Mewis, R. E. y Sutcliffe, O. B. (2020). Classification of fentanyl analogues through principal component analysis (PCA) and hierarchical clustering of GC-MS data. *Forensic Chemistry*, 21, 100287. https://doi.org/10.1016/j.forc.2020.100287 ↑Ver página 47

[Giral *et al.*, 2021] Giral Ramírez, D. A., Montoya Giraldo, O. D., Vargas Robayo, C. Y. y Blanco Valbuena, D. F. (2021). Evaluación de modelos de programación lineal y no lineal para la planeación de sistemas de transmisión en el software GAMS. *Tecnura, 25*(69) [Preprint]. ↑Ver página 44

[Halkidi *et al.*, 2020] Halkidi, M., Batistakis, Y. y Vazirgiannis, M. (2002). Clustering validity checking methods: part II. *ACM Sigmod Record*, *31*(3), 19-27. ↑Ver página 52

[Hernández *et al.*, 2021] Hernández, C., Sánchez Huertas, W. y Gómez, V. (2021). Optimal power flow in electrical energy systems through artificial intelligence techniques. *Tecnura*, *25*(69) [Preprint]. ↑Ver página 44

[Jaimes *et al.*, 2017] Jaimes, B. A., Castro, C. L., Torres, L. B., Silva, G. L. y Braga, A. P. (2017). ClusterCV: *Uma abordagem de visão computacional para a identificação espacial de agrupamentos de dados.* ↑Ver página 45, 49, 53

[Jain *et al.*, 1999] Jain, A. K., Murty, M. N. y Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys (CSUR)*, *31*(3), 264-323. ↑Ver página 44

[Kwon *et al.*, 2018] Kwon, B. C., Eysenbach, B., Verma, J., Ng, K., De Filippi, C., Stewart, W. F. y Perer, A. (2018). Clustervision: Visual supervision of unsupervised clustering. *IEEE Transactions on Visualization and Computer Graphics*, *24*(1), 142-151. ↑Ver página 47

[Lucas *et al.*, 2009] Lucas, F. J., Molina, F. y Toval, A. (2009). A systematic review of UML model consistency management. *Information and Software Technology*, *51*(12), 1631-1645. https://doi.org/10.1016/j.infsof.2009.04.009 ↑Ver página 45

[Luque *et al.*, 2020] Luque Díaz, G. Y., Ramírez Salinas, L. C. y Ruíz Ochoa, M. A. (2020). Fuzzy techniques for environmental impact assessment in hydrocarbons transportation in Colombia. *Tecnura*, *24*(64), 48-65. ↑Ver página 44

[Molina *et al.*, 2010] Molina, J. C. y Torres Moreno, M. E. (2010). Análisis de requerimientos usando BPMN. *Revista Colombiana de Computación*, *11*(1), 85-97. ↑Ver página 45, 50

[Moreno, 2009] Moreno, J. (2009). Trading strategies modeling in Colombian power market using artificial intelligence techniques. *Energy Policy*, *37*(3), 836-843. https://doi.org/10.1016/j.enpol.2008.10.033 ↑Ver página 44

[Mwangi *et al.*, 2014] Mwangi, B., Soares, J. C. y Hasan, K. M. (2014). Visualization and unsupervised predictive clustering of high-dimensional multimodal neuroimaging data. *Journal of Neuroscience Methods*, 236, 19-25. ↑Ver página 47

[Ong *et al.*, 2019] Ong, S. P. (2019). Accelerating materials science with high-throughput computations and machine learning. *Computational Materials Science*, 161, 143-150. https://doi.org/10.1016/j.commatsci.2019.01.013 ↑Ver página 45

[Ramírez-Carvajal *et al.*, 2019] Ramírez-Carvajal, L., Sierra-Peñaranda, G., Puerto-López, K. y Guevara-Ibarra, D. (2019). Computer-aided design software for multi-stage amplifiers with bipo-

lar transistors and field effect. *Technology and Management Journal of Physics: Conference Series*, 1418, 12001. https://doi.org/10.1088/1742-6596/1418/1/012001 ↑Ver página 44

[Ramírez-Escobar *et al.*, 2021] Ramírez-Escobar, C. A. y Buriticá-Arboleda, C. I. (2021). Prototipo de cosecha inteligente de agua lluvia para mejorar la eficiencia energética residencial en Bogotá. *Tecnura, 25*(69) [Preprint]. ↑Ver página 44

[Sánchez-Quintero *et al.*, 2021] Sánchez-Quintero, T., Gómez-Santamaría, C. e Hincapié-Reyes, R. C. (2021). Location estimation of multiple sources based on direction of arrival applying compressed sensing theory. *Tecnura, 25*(67), 40-52. ↑Ver página 44

[Solano Meza *et al.*, 2019] Solano Meza, J. K., Orjuela Yepes, D., Rodrigo-Ilarri, J. y Cassiraga, E. (2019). Predictive analysis of urban waste generation for the city of Bogotá, Colombia, through the implementation of decision trees-based machine learning, support vector machines and artificial neural networks. *Heliyon, 5*(11), e02810. https://doi.org/10.1016/j.heliyon.2019.e02810 ↑Ver página 44

[Tafsast *et al.*, 2017] Tafsast, A., Hadjili, M. L., Bouakaz, A. y Benoudjit, N. (2017). Unsupervised cluster-based method for segmenting biological tumour volume of laryngeal tumours in 18F-FDG-PET images. *IET Image Processing, 11*(6), 389-396. ↑Ver página 47

[Villa Betancur y Giraldo Plaza, 2012] Villa Betancur, A. y Giraldo Plaza, J. E. (2012). Automatización de pruebas unitarias de códigos PHP. *Scientia Et Technica, XVII*(50), 147-151. ↑Ver página 49

[Zhu y Ma, 2018] Zhu, E. y Ma, R. (2018). An effective partitional clustering algorithm based on new clustering validity index. *Applied Soft Computing Journal*, 71, 608-621. https://doi.org/10.1016/j.asoc.2018.07.026 ↑Ver página 47