EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Research in Computing Science

Vol. 152 No. 3
March 2023

Research in Computing Science

Research in Computing Science, Año 22, Volumen 152, No. 3, marzo de 2023, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de marzo de 2023.

Research in Computing Science, year 22, Volume 152, No. 3, March 2023, is published monthly by the Center for Computing Research of IPN.

Volume 152(3)

Advances in Computing Science and Applications

Víctor Reyes-Macedo
Hiram Calvo-Castro (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2023

ISSN: in process

Formerly ISSNs: 1870-4069, 1665-9899

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Ciencia de datos - fundamentos, algoritmos y aplicaciones: Una visión general

Leticia Gómez Rivera, Perfecto Malaquías Quintero Flores,
Sara Gutiérrez Carmona, Miguel Angel Bello Rivera

Tecnológico Nacional de México,
Campus Apizaco, Tlaxcala
México

{m21371205,perfecto.qf,d20371439,m21371204}@apizaco.tecnm.mx

Resumen. En la actualidad, se vive una época en la que los datos son producidos de manera masiva debido a los avances realizados en el tema de comunicaciones y gracias al monitoreo de los procesos de las empresas. Por este motivo, es necesario hablar sobre la ciencia de datos y sobre las ventajas que representa en el análisis de grandes conjuntos de datos para la extracción de información de calidad (smart data) empleados en la toma de decisiones que conduzcan a resultados exitosos. La finalidad de este trabajo es introducir dicho tema, dando en primer lugar, un panorama general del mismo. Para luego, realizar la revisión de conceptos fundamentales de la ciencia de datos. Posteriormente, se presenta una revisión de los algoritmos de aprendizaje máquina más comúnmente utilizados en el proceso de análisis de los grandes datos, específicamente en tareas de regresión, clasificación y agrupamiento. También se incluye la descripción de los procesos planteados por diferentes autores y las debilidades encontradas en ellos, para luego proponer la introducción de un nuevo paso en el proceso de ciencia de datos, que consiste en la definición de objetivos, el cual no se menciona en la literatura aunque es importante para alcanzar los resultados esperados. Finalmente, se describen ejemplos en los que se aplica la ciencia de datos en problemas del mundo real.

Palabras clave: Revisión literaria, ciencia de datos, anomalías de datos, proceso, algoritmos.

Data Science - Fundamentals, Algorithms and Applications: An Overview

Abstract. Nowadays, we live in an era in which data is massively produced due to the advances made in the field of communications and thanks to the monitoring of business processes. For this reason, it is necessary to talk about data science and the advantages it represents in the analysis of large data sets for the extraction of quality information (smart data) used in decision making that leads to successful results. The

purpose of this paper is to introduce this topic, giving first an overview of it. Then, a review of the fundamental concepts of data science is presented. Subsequently, a review of the most commonly used machine learning algorithms in the process of big data analysis, specifically in regression, classification and clustering tasks, is presented. It also includes a description of the processes proposed by different authors and the weaknesses found in them, and then proposes the introduction of a new step in the data science process, which consists in the definition of objectives, which is not mentioned in the literature although it is important to achieve the expected results. Finally, examples are described in which data science is applied to real-world problems.

Keywords: Literature review, data science, data anomalies, process, algorithms.

1. Introducción

De acuerdo con el sitio Go-Globe [1], en el año 2016, en tan solo 60 segundos se realizaban más de 2.3 millones de búsquedas en Google, se generaban más de 3.1 millones de likes, se realizaban más de 3 millones de publicaciones en Facebook y se enviaban más de 150 millones de e-mails, etc. Si se analiza esta información detalladamente, se puede notar que la cantidad de datos que se producen en la actualidad ha incrementado, debido al aislamiento que ha existido durante esta época de pandemia, donde la interacción con otras personas es llevada a cabo mediante herramientas digitales, las cuales almacenan información de las actividades cotidianas. De igual manera, gracias a la optimización de procesos, en las empresas se están empleando herramientas donde a través de sensores, cámaras y sistemas de control se registran los datos que están siendo obtenidos y que representan gran cantidad de información.

Al existir grandes volúmenes de datos, uno de los problemas principales que se presenta es cómo obtener información útil a través de ellos, ya que pueden existir datos irrelevantes que no aporten conocimiento, por lo que solo ocuparán espacio de almacenamiento de forma innecesaria.

Para abordar el reto que representa el contar con grandes cantidades de información se utiliza la ciencia de datos, la cual es un conjunto de métodos que son utilizados para extraer el valor de los datos y también es el descubrimiento del conocimiento, de esta forma el propósito de la ciencia de datos es encontrar estructuras significativas dentro de los datos para poder extraer información de calidad [2], empleando para esto algoritmos y tecnologías de aprendizaje automático, minería de datos, análisis predictivo y visualización de datos.

En [3] mencionan que la ciencia de datos es una teoría y metodología relacionada con la cadena de valor de los datos, cuya tarea principal es almacenar y procesar grandes cantidades de datos, los cuales son heterogéneos, como las imágenes, los videos, textos, etc., lo que los hace más complejos y con mayor nivel de incertidumbre.

La ciencia de datos tiene diversas aplicaciones en la vida cotidiana, como la conservación del medio ambiente a través del análisis de datos en redes sociales

[4], el estudio del rendimiento académico de estudiantes universitarios [5] y la búsqueda de métodos de aprendizaje a través de juegos que sean atractivos para los estudiantes, los cuales son utilizados en escuelas militares y de medicina [6], por mencionar algunas.

Para que la ciencia de datos pueda cumplir con su propósito de procesar los datos, utiliza a la estadística como una base que le permite reducir el ruido presente en ellos, filtrarlos y de esta manera, obtener la información necesaria para extraer conocimiento. Al respecto, en [7] se menciona que la ciencia de datos es una combinación entre la estadística y las ciencias de la computación, debido a que cuando trabajan en conjunto se puede realizar una mejor interpretación de los grandes volúmenes de datos. Por su parte, la estadística se encarga de realizar el análisis de información a través la reducción de dimensionalidad, inferencia, etc. y las ciencias de la computación almacenan la información, la filtran y la preparan para que pueda ser analizada.

A pesar de que los grandes volúmenes de datos proporcionan información relevante sobre la forma en la que se vive, presentan desventajas para comprender cuantitativamente los datos [8], ya que:

- Toman formas complejas.
- Las observaciones que se hacen de los datos no se realizan con un diseño experimental adecuado, por lo que existen sesgos y datos incompletos.
- Desafíos éticos, ya que se puede realizar una identificación personal a través de los datos.
- La transparencia algorítmica.

Contribuciones de este trabajo

El presente trabajo tiene cuatro propósitos principales, los cuales contemplan: El primer propósito es proporcionar una explicación de manera simple del campo de acción de la ciencia de datos e identificar las áreas donde está siendo aplicada, explicando las fuentes de donde provienen los datos y las bases en las que se sustentan tales aplicaciones. El segundo propósito consiste en realizar un análisis de cinco procesos de ciencia de datos, brindando una opinión desde la perspectiva de los autores de este artículo. El tercer propósito de esta revisión es plantear un nuevo paso dentro del proceso de ciencia de datos, el cual destaca que los objetivos de la investigación tienen un papel fundamental en el éxito de su implementación. Y por último, el cuarto propósito consiste en mostrar ejemplos prácticos que detallen claramente los conceptos presentes en el artículo, con la intención de que el lector los relacione con problemas y actividades cotidianas. Con el objetivo de cumplir estos propósitos, el artículo está organizado de la siguiente forma: En la sección dos se hace una introducción a los conceptos fundamentales de la ciencia de datos, la sección tres presenta un análisis sobre la clasificación de la ciencia de datos de acuerdo a los algoritmos de aprendizaje máquina aplicados, en la sección cuatro, se presenta una discusión referente al proceso de la ciencia de datos, la cual concluye con la propuesta de un proceso de ciencia de datos en el que se incluye una fase de definición de objetivos. Y

por último, en la sección cinco, se describe la revisión y análisis de cinco casos de estudio.

2. Conceptos fundamentales

2.1. Tipos de datos

Diariamente se generan grandes cantidades de información que provienen de diferentes fuentes, por ejemplo, el cuerpo humano, las instituciones bancarias, fábricas, oficinas, escuelas, entre otras. Al realizar la recolección de la información, los datos se presentan en diferentes formatos: imágenes, videos, textos, audios, etc., por lo cual, se requiere el uso de métodos estadísticos y de ciencia de datos para lograr extraer conclusiones que ayuden en la toma de decisiones. Uno de los puntos importantes a considerar en el momento de recolección y análisis de datos, es asegurar que no se violen los derechos sociales, profesionales y éticos de la sociedad [9].

A pesar de que existen diversas formas de almacenar los datos, en [10] mencionan que los dos tipos de datos más comunes son los numéricos y categóricos:

- **Datos numéricos**: Están compuestos principalmente por números, por lo que son datos de tipo cuantitativo, es decir, que pueden ser medidos. En este tipo de datos existen dos categorías:
  - Datos continuos: Tienen la característica de contemplar cualquier número de la recta numérica, por lo cual abarcan un conjunto de valores incontable. Algunos ejemplos de estos datos son: el peso de una persona, tiempo de espera en un banco, la temperatura corporal, velocidad de un automóvil, etc.
  - Datos de conteo: En este tipo de datos, solo se contemplan datos de tipo entero. Por ejemplo, el número de alumnos de una clase, número de mesas disponibles en un restaurante, la cantidad de ventanas en una casa, etc.
- **Datos categóricos**: Están constituidos por palabras, símbolos, frases, etc. Son de tipo cualitativo, es decir, datos que se refieren a las características pertenecientes a un objeto y por ese motivo pueden ser divididos en clases. Los datos categóricos pueden dividirse en dos categorías:
  - Datos ordenados u ordinales: La característica principal de este tipo de datos es que siguen un orden inherente. Ejemplo de este tipo de datos están presentes en la descripción de tallas de una prenda: 0-Talla pequeña, 1-Talla mediana, 2-Talla grande, 3-Talla extra grande.
  - Datos no ordenados o categóricos: No tienen un orden que seguir, por lo cual no se puede definir una categoría como anterior a otra. Ejemplos de este tipo de datos son: descripción del clima (lluvioso, templado, etc), la raza de una persona, un tipo de planta, etc.

Conjuntos de datos Cuando los datos son recolectados de manera periódica y con métricas a seguir se forman los **conjuntos de datos**. En estos conjuntos de datos, generalmente la información se agrupa, donde cada fila representa las instancias, ejemplos, registros, objetos, etc. y las columnas representan los atributos, propiedades, características, etc. de esas filas [11]. Un atributo representa una característica del objeto del cual se está obteniendo la información. Un ejemplo que se puede utilizar para describir esto es un auto. El auto representa el objeto o entidad, el conjunto de datos está conformado por los registros de autos que cumplen con cierta métrica, por ejemplo, autos seminuevos, y el color, marca, modelo, etc., representan los atributos de ese auto.

Análisis de datos Una de las partes más importantes en la ciencia de datos es el análisis de la información que está incluida en el conjunto de datos, la cual no se nota a simple vista, pero representa un punto crucial en el proceso de toma de decisiones. En [12] abordan dos enfoques para realizar este análisis, haciendo énfasis en la cantidad de variables:

1. Análisis univariante. En él, se van analizando uno por uno los atributos (variables) que contiene el conjunto de datos. Pueden ser de tipo numérico o categórico. Los atributos categóricos se pueden cambiar a numéricos mediante la codificación, mientras que los atributos numéricos se convierten a categóricos mediante la discretización.
2. Análisis bivariante. En este análisis se utilizan dos atributos, se determina si tienen asociación entre ellos y si es así se busca la fuerza de asociación. En caso contrario se abordan las diferencias existentes entre los dos atributos. Se puede realizar un análisis bivariante de tres maneras diferentes: Con dos atributos numéricos, con dos atributos categóricos y con un atributo numérico y uno categórico.

Por otro lado, en [22], dividen el análisis de la información en cuatro categorías, las cuales abordan el análisis desde una perspectiva en la cual, a través de los datos se pueden realizar predicciones o se puede determinar por qué ocurrieron sucesos específicos:

1. Análisis descriptivo: Este tipo de análisis ayuda a visualizar los acontecimientos que ocurrieron en el pasado con el objetivo de comprender los motivos de fracaso o éxito en ciertas situaciones, para que de esta manera los usuarios interpreten cómo es que esos acontecimientos pueden afectar los resultados futuros. Estos análisis describen los datos, los resumen y permiten que los usuarios puedan comprenderlos de manera simple.
2. Análisis de diagnóstico: La característica principal de este análisis es que permite a los usuarios entender lo que está sucediendo y por qué ocurrió, para que de esta manera se tomen decisiones que ayuden a mejorar lo ocurrido. Analiza los factores que ocasionan un resultado determinado.
3. Análisis predictivo: Este análisis proporciona información acerca de lo que podría ocurrir en el futuro. Su enfoque principal consiste en realizar

predicciones a partir de datos históricos que ayuden a determinar áreas de oportunidad y riesgos.

4. Análisis prescriptivo: No solo se trata de analizar los datos y predecir sucesos futuros, sino que ofrece sugerencias para extraer beneficios y aprovechar las predicciones. El análisis ayuda a optimizar el proceso de toma de decisiones, ya que anticipa qué va a ocurrir, cuándo y por qué razón. Es una guía para los usuarios, en la que infiere cómo le afectarán los hechos y sugiere la opción óptima.

2.2. Anomalías en la información

La información con la que trabaja la ciencia de datos presenta diferentes tipos de anomalías, esto ocurre debido a la naturaleza de los datos, o debido a los instrumentos con los cuales se hace la recolección de datos. Una anomalía es la información que no se encuentra dentro de los patrones normales de los datos.

Las anomalías representan información distinta a la habitual y en la actualidad proporcionan la ventaja de poder detectar sucesos inusuales, como la detección de invasiones en los sistemas y la detección de fraudes, además, ayudan en el análisis de la calidad de los datos, el escaneo de seguridad, el control de sistemas, etc. [13]. Es difícil establecer normas para la detección de anomalías, ya que, toman diferentes formas dependiendo de los tipos de datos que se estén analizando, pero es una tarea importante comprender el tipo de anomalía específica que se presenta en el conjunto de datos con el que se desea trabajar. En [14] se menciona que es importante identificar los tipos de anomalías y sus características en las áreas estadísticas, de ciencia de datos y de aprendizaje máquina, esto con el objetivo de comprenderlas y poder realizar una analítica adecuada de los datos, sin que exista información que altere los resultados obtenidos. Una de las razones fundamentales por la que es importante detectar anomalías en los conjuntos de datos es evitar que los algoritmos se ejecuten mal o que fallen por lo complejos que son los datos del mundo real.

Clasificar los tipos de anomalías es una tarea complicada debido a la variedad que existe por la naturaleza de los datos con los que se está trabajando, pero en [15] dividen a las anomalías en tres categorías:

-Anomalías colectivas: Este tipo de anomalías se forman debido a la combinación de muchos casos, por ejemplo, la secuencia de los datos que se presentan en los sistemas bancarios.

- Anomalías contextuales: Este tipo de anomalías no se distinguen sin la presencia de un contexto. Por ejemplo, un clima caluroso en época invernal mostraría una anomalía, pero si el clima se presenta sin la información de la temporada, se tomaría como un dato válido, sin presencia de anomalías.

- Anomalías puntuales: Son aquellas en donde una sola muestra dentro del conjunto de datos es diferente de las otras muestras.

Al contar con grandes cantidades de datos que se producen cada día y que necesitan ser analizadas para la toma de decisiones, surge el desafío de la detección de anomalías en estos grandes conjuntos de información. La alta dimensionalidad de los datos crea dificultad para la búsqueda de anomalías,

porque aumenta la cantidad de atributos presentes en la información y se necesitan más datos para generalizar estos atributos y detectar los valores atípicos, además, el ruido presente en estos datos afecta la efectividad con la que se puede abordar la búsqueda. Todo esto impacta directamente en las técnicas de detección de anomalías, ya que si se aumentan las dimensiones, se vuelve más complejo el conjunto de datos y aumentan los falsos positivos en la detección [16]. Se debe tener cuidado especial con el tema de las anomalías de los datos, porque gracias a su detección es posible realizar una mejor analítica y toma de decisiones. Si se detectan correctamente las anomalías es posible resolver diversos problemas cotidianos que abarcan desde la seguridad de las grandes empresas, hasta la seguridad en los procedimientos médicos a los cuales se someten tantas personas todos los días.

3. Clasificación de la ciencia de datos desde la perspectiva de algoritmos de aprendizaje automático utilizados

Dependiendo del tipo de datos con los que se trabaje, la ciencia de datos se puede clasificar en dos categorías que son: la ciencia de datos supervisada y no supervisada [2,28]:

- Ciencia de datos supervisada: En esta clasificación, como conjunto de entrenamiento, se utiliza un histórico de datos etiquetados y se busca obtener una función que es utilizada para clasificar datos no etiquetados. La predicción que se realiza tiene el objetivo de clasificar las variables de salida a partir del conjunto de variables de entrada en las que ya se conoce la categoría a la que pertenecen. Este tipo de ciencia de datos necesita suficientes registros etiquetados para que el modelo aprenda a partir de los datos.

- Ciencia de datos no supervisada: Este tipo de clasificación también es conocida como agrupación y se utiliza para definir categorías a partir de la asociación de los datos. El objetivo que persigue es descubrir patrones ocultos dentro de conjuntos de datos no etiquetados.

Por otra parte, en [29] dividen a la ciencia de datos en tres categorías, basadas en el tipo de aprendizaje al que pertenecen los algoritmos que emplea:

- Aprendizaje supervisado: Este tipo de aprendizaje es el que se utiliza con más frecuencia, en él, el programa sabe cuáles son las salidas que va a obtener. Utiliza variables independientes (atributos) para determinar la variable dependiente (clase).

- Aprendizaje no supervisado: Es menos utilizado que el aprendizaje supervisado. En este tipo de aprendizaje no se conocen las clases que existen dentro del conjunto de datos, por lo tanto, la información es agrupada dependiendo de las características que sean similares entre los registros. Este aprendizaje es menos preciso que el aprendizaje supervisado.

- Aprendizaje por refuerzo: En este aprendizaje, se realiza un proceso de entrenamiento de un modelo, el cual consiste en que la computadora interactúe con su entorno incierto y realice acciones. Después se le guía para obtener el resultado que se espera y se le proporcionan recompensas o penalizaciones

dependiendo de las decisiones tomadas. El objetivo de este tipo de aprendizaje es que las recompensas aumenten.

3.1. Algoritmos en la ciencia de datos

Antes de aplicar algún algoritmo de ciencia de datos es importante analizar el problema que se quiere resolver y a partir de ello, realizar un análisis de la naturaleza de los datos, su estructura, los tipos de datos presentes, la cantidad de registros que se tienen, si existen datos ausentes y en qué porcentaje, etc. Es posible implementar los algoritmos de ciencia de datos en cualquier lenguaje de programación, pero es mejor utilizar lenguajes especializados en esta área como lo son R, RapidMiner, Python, SAS Enterprise Miner, etc. [2].

Los algoritmos de la ciencia de datos pueden ser utilizados en diferentes tareas, que comprenden la **clasificación de datos, la regresión, el agrupamiento**, entre otras. Los algoritmos de ciencia de datos mencionados con mayor frecuencia en la literatura [17,18,19,20,21,22,23,24,25,26,27] son:

- **Regresión lineal**: Este tipo de método tiene el objetivo de describir la relación que existe entre una variable dependiente y una o más variables independientes. Las variables independientes y dependientes pueden diferenciarse mediante un supuesto que pertenece al análisis de regresión, en el cual se supone que las variables independientes son exógenas, es decir, que la variable independiente no puede afectarlas y que tampoco existen otras variables fuera del modelo que afecten a la variable dependiente ni a las variables independientes. Un ejemplo de regresión lineal es la predicción del número de ventas en una empresa (variable dependiente) a partir de factores como el costo de envío, el tiempo de entrega del producto y las formas de pago (variables independientes).
  En la Ecuación 1 obtenida de [25] se muestra cómo se calcula la relación lineal entre una variable independiente y una dependiente:

$$y = \beta_0 + \beta_1 x + \varepsilon, \tag{1}$$

  donde:
  - y es la variable dependiente.
  - x es la variable independiente
  - β_0 es la intersección en y o la constante.
  - β_1 es el coeficiente en x o la pendiente.
  - ε es el término de error (el cual refleja que la relación entre x y y no es exacta).
- **Regresión logística**: Este tipo de regresión se basa en la función sigmoidal o logística. Está basada en el principio de probabilidades, el cual se muestra en la ecuación 2 obtenida de [25] y que se define como la probabilidad de que ocurra un suceso (P) dividida por la probabilidad de que no ocurra:

$$probabilidades = \frac{P}{1-P}. \tag{2}$$

La regresión logística expresa el logaritmo natural de las probabilidades como una función lineal de una constante y $k-1$ variables independientes, lo cual se representa con las ecuaciones 3, 4, obtenidas de [25]:

$$\ln\left(\frac{P}{1-P}\right) = \beta_0 + \sum_{i=1}^{k-1} \beta_i x_i, \tag{3}$$

donde P es la probabilidad y β_i representa un cambio en las probabilidades logarítmicas, logrando un cambio en x.

La ecuación 3 puede ser reescrita en términos de las probabilidades P de la siguiente manera:

$$P = \frac{exp(\beta_0 + \sum \beta_i x_i)}{1 + exp(\beta_0 + \sum \beta_i x_i)}. \tag{4}$$

– **K-means**: Es un algoritmo de agrupamiento, por lo que utiliza datos no etiquetados, con el objetivo de encontrar conjuntos (grupos o clases) dentro de esos datos. En [17] mencionan cuatro pasos a seguir en este algoritmo: 1. Inicialización: Generar aleatoriamente los k centroides iniciales (centro de los clústeres). 2. Clasificación: Calcular las distancias de todos los puntos del conjunto a todos los centroides y asignar los datos al centroide más cercano. 3. Calcular los nuevos centroides de los datos. 4. Detener el algoritmo hasta que no se produzcan más cambios. Los pasos 2 y 3 se repiten mientras no se alcance este objetivo. En [12] se presenta la Ecuación 5 para minimizar la varianza total del grupo o la función de error cuadrático:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2, \tag{5}$$

donde:
- J es la función objetivo
- k representa el número de clústeres
- n representa el número de casos
- $x_i^{(j)} - c_j$ es la función de distancia
- $x_i^{(j)}$ es el caso i
- c_j es el centroide para el clúster j

– **K-vecinos más cercanos**: Este tipo de algoritmo es eficaz tanto para la clasificación como para la regresión, pero es más utilizado en tareas de clasificación y predicción. El método trabaja con un conjunto de datos de entrenamiento etiquetado y un conjunto de datos de prueba no etiquetado. Los datos no etiquetados son clasificados en categorías, dependiendo de la cercanía que tengan con sus vecinos. En [21] definen los pasos a seguir para la implementación de este algoritmo: 1. Almacenar el conjunto de datos de entrenamiento. 2. Calcular la distancia Euclidiana con todos los puntos de datos de entrenamiento, para cada dato nuevo no etiquetado. 3.

Encontrar los k-vecinos más cercanos. 4. Asignar el punto no etiquetado a la clase que contenga la mayor cantidad de vecinos más cercanos. 5. Repetir el procedimiento hasta asignar cada punto no etiquetado a su clase correspondiente.

En la Ecuación 6 obtenida de [12] se muestra cómo se calcula la distancia Euclidiana, que es la más utilizada para determinar la etiqueta del dato a partir de los vecinos más cercanos:

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}, \tag{6}$$

– **Naive Bayes:** Es un método basado en el Teorema de Bayes, el cual define una ecuación que describe la probabilidad de que ocurra un evento dada la probabilidad de eventos relacionados. La característica vuelve *ingenuo* a este algoritmo, es que asume que las variables del conjunto son independientes entre ellas, es decir, que la aparición de una variable no tiene nada que ver con las demás. Naive Bayes tiene tres variaciones para su clasificador:
  1. Bernoulli: Para conjuntos de datos binarios.
  2. Multinomial: Para conjuntos de datos discretos.
  3. Gaussiano: Para conjuntos de datos que se ajustan a una distribución normal.

  El Teorema de Bayes se muestra en la Ecuación 7 [27]:

$$P(L|caracteristicas) = \frac{P(caracteristicas|L) \times P(L)}{P(caracteristicas)}, \tag{7}$$

  donde:
  - $P(L)$ es la probabilidad de L antes de que se observen los datos.
  - $P(L|caracteristicas)$ es la probabilidad que se quiere calcular, la probabilidad de L dadas las $caracteristicas$.
  - $P(caracteristicas|L)$ es la probabilidad de las $caracteristicas$ dada una etiqueta L.
  - $P(caracteristicas)$ Es la probabilidad de las $caracteristicas$.

– **Árboles de decisión:** Este tipo de algoritmo se usa tanto para realizar clasificación como regresión. Los árboles de decisión están compuestos de nodos y ramas. Los nodos están conformados por las características de una categoría a clasificar y las ramas representan los valores que puede tomar el atributo. Este algoritmo pretende dividir el conjunto de datos en subconjuntos más pequeños.
  Si se utilizan los árboles de decisión para la clasificación es necesario calcular su *Entropía* y *Ganancia*. En [20], mencionan que la entropía se emplea para medir el grado de aletoriedad que existe en el conjunto de datos. El valor de la entropía se mide entre 0 y 1, siendo 0 el resultado esperado y 1 el

peor resultado que se puede obtener en el cálculo. Por otro lado, la ganancia de información es una métrica que informa de manera intuitiva sobre el conocimiento del valor de una variable aleatoria. Al contrario de la entropía, mientras mayor sea el valor de la ganancia de información, mejor.
Las Ecuaciones 8 y 9 fueron obtenidas de [12]. La Ecuación 8 se utiliza para calcular la entropía:

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i, \tag{8}$$

donde:
- $E(S)$ representa la entropía de un atributo
- p_i representa la probabilidad del i-ésimo valor del atributo
Después, se divide el conjunto de datos en sus atributos y se calcula la entropía para cada uno.
Para obtener la ganancia de información se utiliza la Ecuación 9:

$$Ganancia(T, X) = Entropia(T) - Entropia(T, X), \tag{9}$$

donde:
- T es entropía antes de dividir el conjunto de datos.
- X es la entropía total de la división utilizando un atributo específico.

– **Máquinas de Soporte Vectorial:** Este tipo de método es ocupado para realizar clasificación lineal, no lineal, detección de valores atípicos y regresión. El objetivo de este algoritmo es encontrar un hiperplano con el que se separen los puntos de un vector en dos clases. Los puntos de datos cercanos al hiperplano son denominados vectores de soporte. Las principales limitaciones de este algoritmo son la velocidad y el tamaño de los datos, ya que no es adecuada en la clasificación de grandes conjuntos. Las ecuaciones 10 y 11 obtenidas de [23] tienen el objetivo de encontrar la mejor forma de separar el hiperplano:

$$wx - b = 0, \tag{10}$$

donde:
- w es un vector de valores reales
- x es un vector de características de entrada
- b es un número real
- wx representa $w^{(1)}x^{(1)} + w^{(2)}x^{(2)} + \cdots + w^{(D)}x^{(D)}$ y D representa el número de dimensiones del vector x

Para definir dos hiperplanos paralelos se hace uso de las siguientes ecuaciones:

$$\begin{aligned} wx - b &= 1, \\ wx - b &= -1. \end{aligned} \tag{11}$$

– **Bosques aleatorios:** Este tipo de algoritmo es utilizado para realizar predicciones y es adecuado para conjuntos de datos medianos y grandes. En

un bosque aleatorio se construyen muchos árboles de decisión individuales y después se promedian las predicciones realizadas por cada uno de ellos. Aunque los árboles de decisión son más fáciles de interpretar, los bosques aleatorios tienen mejores resultados realizando predicciones. Para controlar la profundidad de los árboles en un bosque aleatorio, es necesario definir desde la selección del modelo, el tamaño del subconjunto de variables predictoras [24].

– **DBSCAN:** Es uno de los algoritmos de agrupamiento más utilizados. Está enfocado en encontrar áreas de alta densidad en el espacio de distribución de los datos. Para medir la densidad es necesario tomar un punto del conjunto y encontrar los puntos más cercanos a él (formando una vecindad), haciendo uso de una métrica de distancia. Mientras más puntos se encuentren dentro de la vecindad, mayor densidad tendrá el clúster. Este algoritmo se emplea de forma recursiva, eligiendo un punto y verificando sus puntos vecinos para establecer la densidad.

  Para emplear el algoritmo DBSCAN es necesario tomar en cuenta dos valores [22]:
  - *epsilon*: Es un número positivo que sirve como métrica para medir la distancia máxima entre los dos puntos del clúster.
  - $MinPts$: Es un número natural que define un umbral mínimo para establecer un área de puntos como densa. Este parámetro es definido por el usuario.

4. Proceso de la ciencia de datos

En la ciencia de datos, se lleva a cabo un proceso para eliminar la información irrelevante en los conjuntos de datos, obtener la información importante de estos conjuntos y conocer la naturaleza de los registros con los que se está trabajando.

En [2] mencionan que el paso más importante para comenzar un proceso de ciencia de datos es la necesidad de analizar un problema, ya que sin una buena definición del problema no es posible aplicar la ciencia de datos. El proceso que proponen para llevar a cabo un proyecto de ciencia de datos consta de cinco pasos: 1. Obtener conocimiento previo del problema para conocerlo a profundidad, además de entender los datos relacionados a ese problema. 2. Preparar los datos, es decir, ajustar los datos de modo que se presenten en la forma requerida por los algoritmos de ciencia de datos (sin valores perdidos, con selección de características, sin anomalías en los datos, etc.). 3. Realizar la aplicación del modelo, es decir, representar los datos y las relaciones existentes entre ellos. 4. Integrar el modelo a su entorno de producción, aquí es donde se evalúa el modelo, el tiempo en el que responde a lo que se le solicita y el mantenimiento que requiere. 5. Obtener conocimiento acerca de los datos analizados, después de haber realizado la extracción de información no trivial a partir del conjunto de datos.

Por otro lado, en [30], señalan que la ciencia de la datos está relacionada con la gestión del conocimiento, por este motivo, proponen implementar los proyectos de ciencia de datos en el contexto del proceso del conocimiento, el cual está dividido en 5 pasos: 1. Establecer sistemas de información e infraestructura en donde se generen los datos. 2. Recopilar los datos de las fuentes establecidas, adquirir la información necesaria para llevar a cabo el proceso y realizar gestión del contenido. 3. Procesar, analizar y tratar la información/datos con el objetivo de reducir datos irrelevantes y extraer la información útil del conjunto. 4. Gestionar la información y compartir el conocimiento obtenido del análisis. 5. Usar la información y el conocimiento, aplicándolo en las áreas necesarias.

En el trabajo [31], proponen un entorno de trabajo para realizar el proceso de ciencia de datos, el cual consta de 8 pasos: 1. Especificar el problema que se necesita resolver, conociendo el dominio en el que se desarrolla el problema. 2. Descubrir los datos, es decir, buscar fuentes de datos ya existentes que estén relacionadas con el problema, antes de realizar una nueva recopilación de datos. 3. Establecer el cumplimiento de normas referentes al acceso, la difusión y la destrucción de los datos e introducir la información en plataformas de gestión. 4. Realizar la gestión de los datos donde se evalúa la calidad, la preparación y vinculación de la información. 5. Evaluar la idoneidad, este paso abarca desde tabulaciones y visualizaciones descriptivas hasta análisis complejos de los datos, y debe caracterizar el contenido informativo de los resultados. 6. Modelizar y realizar análisis estadísticos, lo cual es fundamental para obtener conclusiones sólidas obtenidas a partir de información incompleta. 7. Comunicar y difundir, es decir, compartir los datos, código que fue desarrollado, documentos referentes al trabajo y realizar presentaciones, conferencias, publicaciones, etc. 8. Realizar una revisión ética, proporcionando un conjunto de principios, en donde se incluyan consideraciones referentes a la vigilancia masiva, privacidad y soberanía de los datos.

Por otra parte, en el trabajo [32], proponen un método genérico a seguir en un proyecto de ciencia de datos, el cual fue obtenido después de analizar más de 150 actividades de ciencia de datos. El método consta de 10 pasos que son: 1. Identificar el problema o el fenómeno que requiere ser investigado y establecer cuál es el resultado que se espera. 2. Definir el problema utilizando el conocimiento del dominio, identificando factores críticos a analizar. 3. Formular la hipótesis a evaluar sobre los parámetros y modelos. 4. Diseñar el análisis desde el descubrimiento y adquisión de datos hasta el análisis e interpretación de resultados. 5. Garantizar la validez conceptual del diseño del análisis de datos. 6. Diseñar, probar y evaluar cada paso, seleccionando la clase de algoritmos pertinentes para la preparación de los datos y modelos, seleccionando y ajustando los algoritmos para satisfacer los requerimientos analíticos y garantizando la validez de la aplicación del análisis de datos. 7. Ejecutar una segmentación encauzada, asegurando que se cumplen los requerimientos. 8. Garantizar la validez de los resultados con respecto al problema investigado. 9. Interpretar los resultados con respecto a los modelos, métodos y requerimientos del análisis, y

evaluar los resultados. 10. Poner en funcionamiento y supervisar la segmentación encauzada y sus resultados.

A diferencia de los trabajos mencionados anteriormente, en [33] dividen el proceso de la ciencia de datos en nueve etapas, las cuales se presentan como una extensión del ciclo de vida de los datos. Las etapas son: 1. Realizar un diseño experimental, 2. Obtener o generar los datos y construir los modelos de datos, 3. Generar la hipótesis y explorar los datos, 4. Realizar una limpieza y organización de los datos, 5. Preparar los datos (valores perdidos y selección de características), 6. Realizar una estimación del modelo, 7. Llevar a cabo la simulación del algoritmo con los datos, 8. Visualizar los resultados obtenidos y por último, 9. Publicar el manuscrito del artefacto para que se pueda reusar y se puedan reproducir los resultados obtenidos durante el proceso.

Observando los pasos que se siguen en cada uno de los procesos de ciencia de datos que se han descrito, se pueden identificar algunos puntos débiles que ocasionarían que el proyecto no finalice con éxito, por ejemplo, en el trabajo [2] no se menciona un paso en el que se recolecten o generen los datos, a pesar de que el paso 1 contempla entender los datos involucrados en el problema. Otro de los puntos débiles encontrados en este proceso es que no incluye un paso donde se realice una publicación o un reporte de los hallazgos obtenidos, el cual es el producto final del proceso. Por otra parte, en el trabajo [30], a pesar de que el primer paso contempla establecer sistemas de información e infraestructura para la generación de datos, no se establece un paso anterior en donde se realice la observación del entorno para identificar un problema a resolver. Otro punto débil encontrado en el proceso es que, aunque en el paso 3 se hace referencia al análisis y tratamiento de los datos, no se refleja la implementación de algoritmos de ciencia de datos. En el trabajo [31], mencionan el uso de métodos estadísticos que ayuden a extraer información relevante a partir de los datos, pero en ninguno de sus pasos se establece la implementación de los algoritmos correspondientes al área de ciencia de datos. Otro de los puntos débiles identificados en este proceso, es que no se establece un paso en donde se haga una evaluación de los algoritmos implementados a lo largo del proceso, el cual es fundamental para medir el grado de precisión de los resultados obtenidos. Por otro lado, en el trabajo [32], el punto débil que se identificó, fue que no se establece un paso de publicación, el cual es importante para que los resultados obtenidos puedan ser validados y reproducidos por otros investigadores o personas interesadas en el tema. Por último, en el trabajo [33], el paso 1 menciona que se debe realizar un diseño experimental, pero al no contar con un paso anterior en donde se observe el entorno y se identifique el problema, no es posible la realización de dicho diseño.

Debido a los puntos débiles identificados en los cinco procesos de ciencia de datos analizados, es necesario establecer un proceso que contemple las etapas fundamentales para el desarrollo exitoso del proyecto, además, es necesario incluir un paso que no fue encontrado en los procesos anteriores y que representa la dirección a seguir en el proyecto de ciencia de datos que se necesita implementar: definir objetivos, ya que sin ellos, el desarrollo del proyecto de

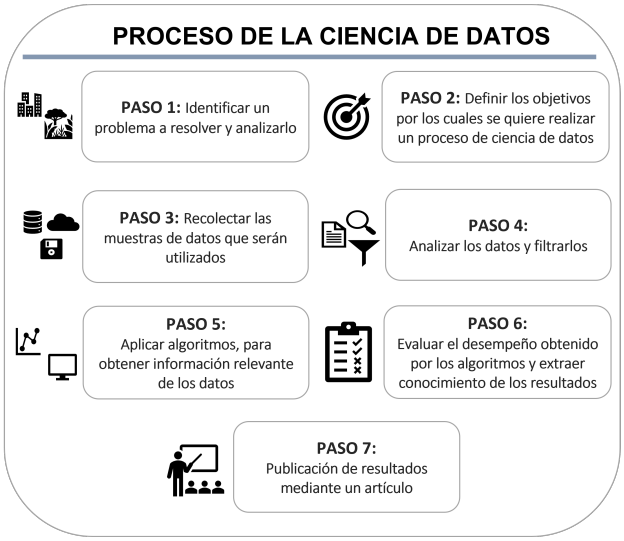


Fig. 1. Pasos del proceso a seguir en un proyecto de ciencia de datos. *Esta imagen es de autoría propia, en ella se contemplan los pasos más relevantes de los cinco procesos analizados, agregando la propuesta del Paso 2.

ciencia de datos no contempla una base sólida en la que apoyarse y sobre la cual dirigirse, lo que conlleva a que no se evalúe de manera adecuada el progreso del proyecto, ni los resultados obtenidos. Por ese motivo, en este artículo, se propone introducir al proceso de ciencia de datos una fase de definición de objetivos, y se enumeran los pasos fundamentales para desarrollar el proyecto, los cuales se obtuvieron de los cinco procesos analizados, esto se detalla en la Figura 1. El proceso consta de siete pasos que se describen a continuación:

- Paso 1. Identificar un problema a resolver y analizarlo, fundamentando el empleo de ciencia de datos en el proyecto: Este paso implica observar el entorno que nos rodea, ya sea en un entorno natural o social e identificar algún problema que esté conformado por grandes cantidades de datos, de las cuales sea necesario obtener información. Se analiza el problema, los atributos que lo conforman y se obtiene toda la información posible relacionada con él. En este paso es importante observar si el problema necesita resolverse mediante técnicas de ciencia de datos.
- Paso 2. Definir los objetivos, por los cuales, se requiere realizar un proceso empleando ciencia de datos: Con los objetivos se determina qué es lo que se quiere lograr al realizar el desarrollo del proyecto de ciencia de datos. Por ejemplo, si se realiza ciencia de datos sobre información de una empresa dedicada a las ventas, se puede establecer el objetivo de analizar los datos de sus productos en busca de defectos en su maquinaría o en el material que utiliza; o bien, se puede analizar qué productos son los que más se venden juntos para armar paquetes de venta.

- Paso 3. Recolectar las muestras de datos que serán utilizadas: Estas muestras pueden ser generadas o recolectadas del entorno en el cual haya sido identificado el problema, para esto, se necesita seguir una regla de recolección de atributos por cada registro.
- Paso 4. Analizar los datos y filtrarlos: En esta etapa, se emplean técnicas estadísticas para analizar los conjuntos de datos (obtener la media, mínimo y máximo de los datos, desviación estándar, etc.) que se obtuvieron en el paso anterior, también se filtran los datos, tomando en cuenta los datos ausentes en el conjunto.
- Paso 5. Aplicar algoritmos para obtener información relevante: Este paso consiste en aplicar algoritmos estadísticos o de aprendizaje máquina para lograr clasificar los datos o si es el caso, realizar la predicción de los resultados que serán obtenidos.
- Paso 6. Evaluar el desempeño obtenido por los algoritmos, extrayendo el conocimiento de los resultados: Se evalúa el grado de precisión con el que los algoritmos manejaron los conjuntos de datos para realizar las tareas asignadas y se analizan los resultados para obtener información relevante para la toma de decisiones.
- Paso 7. Publicación de resultados mediante un artículo: Este paso es importante para poder comunicar los resultados obtenidos a lo largo del proceso.

A pesar de que en esta sección se han descrito diferentes pasos en el proceso de la ciencia de datos, es importante identificar que el punto de partida es definir un problema que puede ser resuelto por medio de ciencia de datos, y al final, la meta es extraer información relevante sobre los datos analizados, los pasos intermedios pueden variar dependiendo de los tipos de datos que se quieran analizar y de las métricas para la obtención de información y de divulgación de resultados.

5. Ejemplos de aplicación de ciencia de datos

En esta sección, se presentan tres ejemplos de aplicaciones de la ciencia de datos en diversas áreas. Estas aplicaciones ayudan a obtener información útil e identifican aspectos del problema, que no hubieran sido detectados sin el análisis de los datos.

La ciencia de datos se puede aplicar en el área deportiva, realizando el monitoreo del estado en el que se encuentra un atleta. En [34] analizaron la actividad cerebral de atletas de tiro con arco, utilizando pruebas de electroencefalograma (EEG) y el algoritmo de Bosques Aleatorios. El propósito de realizar el monitoreo fue permitir a los entrenadores conocer el estado de los atletas antes de la competencia, ya que la carga psicológica, el control de los nervios, la toma de decisiones y la respuesta rápida son puntos importantes a considerar para que los atletas tengan éxito. A través de los datos extraídos de la prueba de EEC (como el índice de estado funcional cerebral, la entropía, los

neurotransmisores, etc.) clasificaron el estado competitivo de los atletas en cinco categorías: excelente, bueno, general, malo y muy malo. La precisión después de aplicar el algoritmo de Bosques Aleatorios fue del 89.74 %, la cual fue comparada con la precisión obtenida mediante modelos de Máquinas de Soporte Vectorial, que presentaron un rendimiento más bajo, con un 80.35 % de precisión.

Otro de los ejemplos en los que se puede aplicar la ciencia de datos es en [35], en este trabajo, los autores proponen un modelo que evalúa la calidad de enseñanza en un aula invertida. En este tipo de aulas, se utiliza un método de enseñanza en donde el estudiante es el pilar más importante, por lo cual, evalúa la calidad de la enseñanza que está recibiendo y también factores clave en su correcto aprendizaje. En este ejemplo, utilizaron el algoritmo de Máquinas de Soporte Vectorial enfocado en la regresión y seleccionaron 4000 grupos de datos para el entrenamiento del algoritmo y 500 grupos de datos para pruebas experimentales. Los datos utilizados como entrada del algoritmo son los indicadores de evaluación de expertos, docentes y estudiantes. Los indicadores contemplan una evaluación para la planificación docente, actitud docente, los dispositivos de enseñanza, entorno de enseñanza y la situación del aula. Después de que aplicaron el algoritmo de Máquinas de Soporte Vectorial analizaron cuatro aspectos: la transformación de los métodos de enseñanza, la abundancia de recursos didácticos, el cambio de iniciativa de aprendizaje de los estudiantes y la evaluación de calidad de la enseñanza en el aula invertida, obteniendo una precisión superior al 99.70 % y un error máximo de 0.04.

La ciencia de datos también puede emplearse en el área médica. En este ejemplo se describe cómo es que la ciencia de datos puede ayudar en el diagnóstico oportuno del cáncer oral. En [36] utilizan aprendizaje automático para identificar el cáncer oral por medio de imágenes, el cual es un método no invasivo y cómodo para los pacientes. Actualmente, se puede aplicar la ciencia de datos en información médica debido a que en estos años se han ido digitalizando los expedientes y análisis médicos. El cáncer oral es uno de los cánceres con tasas más altas de mortalidad, por lo cual su diagnóstico en etapas tempranas es importante, y para eso se necesita identificar las lesiones que se pueden transformar en malignas. En el ejemplo, la clasificación del cáncer se realizó con diferentes métodos de ciencia de datos: las Máquinas de Soporte Vectorial tuvieron una precisión de 82 % en la detección de mucosa normal y patológica. Por otra parte, los árboles de clasificación RepTree y J48Tree, tuvieron un 78.7 % de precisión en la detección de cáncer oral.

En el trabajo [37], se muestra un ejemplo en el cual se utilizan algoritmos de aprendizaje supervisado con el objetivo de predecir la presencia de COVID-19 en una persona. Este virus ha afectado a muchas personas a lo largo del mundo, ya que daña a los órganos del cuerpo debido a la inflamación generalizada que provoca. En este trabajo utilizan un conjunto de datos públicos que se encuentra en el sitio web Kagle, denominado *Síntomas y presencia de COVID-19*, el cual contiene 5434 registros con 20 características, entre las que se incluyen: fiebre, tos seca, dolor de cabeza, fatiga, hipertensión, etc. Los algoritmos de

Tabla 1. Valores de las métricas por cada algoritmo implementado.

Algoritmo	Precisión	Instancias bien clasificadas	Instancias mal clasificadas	Estadística Kappa	Error absoluto medio	Tiempo (s)
J48 DT	0.986	4144	60	0.972	0.024	0.03
RF	0.988	4154	50	0.976	0.023	0.18
SVM	0.988	4154	50	0.976	0.012	3.12
KNN	0.987	4149	55	0.973	0.022	0.01

Tabla 2. Resultados de Sensibilidad y Especificidad obtenidos por los algoritmos

Algoritmo	Sensibilidad	Especificidad
CNN	0.79	0.64
SVM	0.8	0.71
KNN	0.75	0.63
NBC	0.73	0.63

aprendizaje supervisado que utilizaron fueron: Árbol de decisión J48 (J48 DT), Bosques Aleatorios (RF), Máquinas de Soporte Vectorial (SVM) y K-Vecinos más Cercanos (KNN), implementados en la herramienta WEKA. En este ejemplo, utilizaron seis métricas para medir el rendimiento de los algoritmos: 1. Máxima precisión. 2. Mayor cantidad de instancias clasificadas correctamente. 3. Menor cantidad de instancias clasificadas incorrectamente. 4. Estadística Kappa (determina la confiabilidad de los resultados entre dos evaluaciones sobre la misma instancia). 5. Error Absoluto Medio más bajo y 6. Menor tiempo necesario para construir el modelo. Los resultados obtenidos para estas métricas se muestran en la Tabla 1, donde se puede visualizar que los algoritmos con mejor rendimiento al realizar la clasificación fueron Bosques Aleatorios y Máquinas de Soporte Vectorial, aunque el algoritmo con menor tiempo de implementación fue el de K-Vecinos más Cercanos.

La ciencia de datos también puede aplicarse en el desarrollo de herramientas que permitan el diagnóstico de la oncología mamaria. En [38], muestran un ejemplo en donde hacen uso de algoritmos de ciencia de datos para diagnosticar el cáncer de mama, a partir de un conjunto de imágenes que muestran la temperatura dentro de los tejidos y órganos. Los datos que se utilizaron para entrenar y probar los algoritmos fueron exámenes médicos reales de radiometría de microondas, en donde se incluyó la información de 302 pacientes, de los cuales 124 tenían un diagnóstico de cáncer. Las características de cada uno de esos registros incluían datos como: las temperaturas medidas, edad, temperatura ambiente, tamaño de los senos, diagnóstico, etc. Los algoritmos de ciencia de datos que fueron utilizados en este ejemplo contemplaban redes neuronales convolucionales (CNN), Máquinas de Soporte Vectorial (SVM), K-Vecinos más Cercanos (KNN) y el Clasificador Naive Bayes (NBC), los cuales mostraron el desempeño que se visualiza en la Tabla 2.

Como puede notarse, el algoritmo de Máquinas de Soporte Vectorial es el que obtuvo los mejores resultados en comparación con los otros tres algoritmos.

Incorporar algoritmos de ciencia de datos en la medicina es importante para que los médicos, en conjunto con los sistemas informáticos realicen mejores diagnósticos que contribuyan a mejorar la vida de las personas.

6. Conclusiones

En este artículo se mostraron algunos de los conceptos clave que involucra la ciencia de datos. Al ser un área tan amplia no se abordaron todas las tareas que se pueden realizar con ella, como la implementación de motores de recomendación o la detección de anomalías, etc., y tampoco los algoritmos empleados en estas tareas. Sin embargo, en este artículo se abordaron temas importantes que ayudan a comprender con mayor facilidad cómo es que la ciencia de datos y sus algoritmos están presentes en nuestro entorno real y cómo es que con acciones que son parte de nuestra rutina se generan grandes cantidades de información, que al ser analizadas adecuadamente pueden contribuir a mejorar la forma en que vivimos. En este artículo, se realizó un análisis de los conceptos clave que se manejan en el área de ciencia de datos, expresado desde el punto de vista de los autores de las fuentes bibliográficas estudiadas y de los autores del presente artículo.

Además, se dedicó una sección específica al proceso que se sigue en el desarrollo de proyectos de ciencia de datos, debido a que este proceso es primordial para obtener los resultados esperados al finalizar la implementación. Dentro de la misma sección, se tomaron algunos de los pasos de los procesos analizados y se estructuraron de tal manera que contemplaran las etapas elementales para el desarrollo exitoso del proyecto. También, se incluyó un paso que no fue identificado en los procesos analizados: Definir los objetivos. Este paso es fundamental en el desarrollo de un proyecto de ciencia de datos porque está enfocado en resolver un problema específico, pero puede ser abordado mediante diversas técnicas y algoritmos que producirían resultados diferentes. Por ejemplo, si uno de los objetivos es mejorar el tiempo en el que se obtienen resultados con un algoritmo, se elegirá el que sea más rápido, por el contrario, si el objetivo está enfocado en obtener un mayor porcentaje de rendimiento que se vea reflejado en las métricas de desempeño, entonces el método debe cambiar por el que produzca mejores resultados, por este motivo es importante definir desde el inicio el problema que se quiere resolver, pero también los objetivos que quieren ser alcanzados al finalizar el proyecto.

Por último, en este artículo se describieron algunas de las aplicaciones en las que se describen problemas analizados desde un enfoque de ciencia de datos, en los cuales se mencionan los datos utilizados, los algoritmos de aprendizaje máquina empleados y los resultados obtenidos. La ciencia de datos es una poderosa herramienta que sirve para mejorar aspectos de nuestro entorno que de otra manera tardarían mucho tiempo y utilizarían muchos recursos para ser resueltos, por esta razón, la investigación y desarrollo en esta área es importante

para enfrentar los problemas que se presentan en la actualidad y que involucran una gran cantidad de datos producidos de manera masiva.

Referencias

1. GO-GLOBE: Things That Happen Every 60 Seconds [Infographic]. https://www.go-globe.com/60-seconds/ (2022)
2. Kotu, V., Deshpande, B.: Data Science Concepts and Practice. 2nd edn. Morgan Kaufmann Publishers, U.S. (2019)
3. Xu, Z., Tang, N., Xu, C., Cheng, X.: Data science: connotation, methods, technologies, and development. Data Science and Management 1, 32–37, (2021)
4. Toivonen, T. et al.: Social media data for conservation science: A methodological overview. Biological Conservation 23, 298–315, (2019)
5. Padilla, V., Morales, S., Quintana, M., Flores, J., Herrera, O.: Reducción de la dimensión de registros de evaluaciones académicas aplicando el algoritmo K-means. Research in Computing Science 148(7), 515–526, (2019)
6. Alonso, C., Calvo, A., Freire M., Martínez, I., Fernández, B.: Applications of data science to game learning analytics data: A systematic literature review. Computers & Education 141, (2019)
7. Ceri, S.: On the role of statistics in the era of big data: A computer science perspective. Statistics and Probability Letters 136, 68–72, (2018)
8. Olhede, S., Wolfe, P.: The future of statistics and data science. Statistics and Probability Letters 136, 46–50, (2018)
9. Muqeeth, M., Kolhar, M., Al Ameen, A., Rahmath, M.: Data Science Techniques, Tools and Predictions. International Journal Of Recent Technology And Engineering (IJRTE), 8(6), 5661-5668, (2020)
10. Gutman, A., Goldmeier, J.: Becoming a Data Head, 1st edn. John Wiley & Sons, Inc., Indiana (2021)
11. Zaki, M., Meira, W.: Data Mining and Machine Learning, 2nd edn. Cambridge University Press, UK (2020)
12. Saedsayad.com: Data Mining Map. https://www.saedsayad.com/data_mining_map.htm. Last accessed 2 Feb 2022
13. Foorthuis, R.:On the nature and types of anomalies: a review of deviations in data. International Journal of Data Science and Analytics 12, 297–331, (2021)
14. Foorthuis, R.: A Typology of Data Anomalies. In IPMU, International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 26-38. Cádiz, Spain (2018)
15. Sangaiah A., Zhang, Z., Sheng, Q.: Computational intelligence for multimedia big data on the cloud with engineering applications, 1st edn. Academic Press, London (2018)
16. Thudumu, S., Branch, P., Jin, J., Singh, J.: A comprehensive survey of anomaly detection techniques for high dimensional big data. Journal of Big Data 7(42), (2020)
17. Ortega, J., Almanza, N., Vega, A., Pazos, R., Zavala, J., Martínez, A.: The K-Means Algorithm Evolution. 10.5772/intechopen.85447 (2019)
18. Berry, M., Mohamed, A., Yap, B.: Supervised and Unsupervised Learning for Data Science. Springer Nature, Switzerland (2020)
19. Kroese, D., Botev, Z., Taimre, T., Vaisman, R.: Data Science and Machine Learning: Mathematical and Statistical Methods. CRC Press, Boca Raton (2019)

20. Jijo, B., Mohsin, A.: Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends. 2. pp. 20-28, (2021)
21. Taunk, K., De, S., Verma, S., Swetapadma, A.: A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. 2019 International Conference on Intelligent Computing and Control Systems (ICCS), pp. 1255-1260 (2019)
22. Balusamy, B., Abirami, N., Kadry, S., Gandomi, A.: Big Data: Concepts, Technology, and Architecture, 1st edn. John Wiley & Sons, Inc., USA (2021)
23. Burkov, A.: The Hundred-Page Machine Learning Book. Andriy Burkov (2019)
24. Schonlau, M., Zou, R.: The random forest algorithm for statistical learning. The Stata Journal, 20(1), 3–29, (2020)
25. Daniels, L., Minot, N.: An Introduction to Statistics and Data Analysis Using Stata. 1st edn. SAGE, California (2018)
26. Taulli, T.: Artificial Intelligence Basics. 1st. edn. Apress, USA (2019)
27. Voron, F.: Building Data Science Applications with FastAPI. 1st. edn. Packt Publishing Ltd., UK (2021)
28. Ghavami, P.: Big Data Management, 1st edn. De Gruyter, Berlin/Boston (2020)
29. Qamar, U., Summair M.: Data Science Concepts and Techniques with Applications. Springer Nature, Singapore (2020)
30. Chen, J., Ayala, B., Alsmadi, D., Wang, G.: Fundamentals of Data Science for Future Data Scientists. Analytics And Knowledge Management, 167–194 (2018)
31. Keller, S., Shipp, S., Schroeder, A., Korkmaz, G.: Doing Data Science: A Framework and Case Study. Harvard Data Science Review, 2(1), (2020)
32. Braschler, M., Stadelmann, T., Stockinger, K.: Applied Data Science. 1st edn. Springer, Switzerland (2019)
33. Stodden, V.: The data science life cycle: a disciplined approach to advancing data science as a science. Communications of the ACM 63(7), 58–66 (2020)
34. Li, X.: Athletes' State Monitoring under Data Mining and Random Forest. Journal of Sensors 2022, 1–11 (2022)
35. Fu, J., Li, J.: Teaching Quality Evaluation Model of a Flipped Classroom in Colleges and Universities Based on Support Vector Machine. Wireless Communications and Mobile Computing 2022, 1–12 (2022)
36. García-Pola, M., Pons-Fuster, E., Suárez-Fernández, C., Seoane-Romero, J., Romero-Méndez, A., López-Jornet, P.: Role of Artificial Intelligence in the Early Diagnosis of Oral Cancer. A Scoping Review. Cancers 13(18), 4600 (2021)
37. Villavicencio, C., Macrohon, J., Inbaraj, X., Jeng, J., Hsieh, J.: COVID-19 Prediction Applying Supervised Machine Learning Algorithms with Comparative Analysis Using WEKA. Algorithms, 14, 201 (2021)
38. Khoperskov, A., Polyakov, M.: Improving the Efficiency of Oncological Diagnosis of the Breast Based on the Combined Use of Simulation Modeling and Artificial Intelligence Algorithms. Algorithms, 15, 292 (2022)

ISSN 1870-4069

Trazabilidad de imágenes digitales usando Blockchain

José Antonio Jiménez Miramontes[1], Rocío Aldeco-Pérez[2]

[1] Universidad Nacional Autonoma de México,
IIMAS, Posgrado en Ciencias e Ingeniería de la Computación,
México

[2] Universidad Nacional Autonoma de México,
Facultad de Ingeniería,
México

ja.jimenez.mi@gmail.com
raldeco@unam.mx

Resumen. El control que la gente tiene sobre sus activos digitales una vez que están en Internet es debatible. Si nos enfocamos en imágenes que se comparten en Internet vemos casos en donde imágenes públicas son editadas sin el consentimiento del dueño o imágenes compartidas de manera confidencial que son compartidas a terceros sin el consentimiento del dueño generando daños personales o siendo parte del fenómeno de *fake news*. Una solución a esta problemática es el tener la historia de lo que ha sucedido a dicha imagen como evidencia de las operaciones realizadas y de ahí tomar acciones que pueden ir desde lo personal hasta lo legal. Esta historia o "linaje electrónico" de la imagen debe ser confiable, de ahí proponemos el uso de Blockchain. Sin embargo, las Blockchain públicas más usadas son lentas y consumen altas cantidades de energía, además de no ser eficientes cuando de almacenamiento de imágenes se trata. En vista de esto se propone el uso de una Blockchain llamada Solana junto con el protocolo IPFS para el almacenamiento de imágenes. Como consecuencia se obtendrá un esquema distribuido, eficiente y confiable que posteriormente puede ser usado para dar trazabilidad a diversos activos digitales.

Palabras clave: Blockchain, trazabilidad, cifrado simétrico.

Digital Image Traceability Using Blockchain

Abstract. The control that people have over their digital assets once they are on the Internet is debatable. When focusing on images shared online, we can observe cases where public images are edited without the owner's consent, or confidentially shared images are distributed to third parties without authorization, causing personal harm or contributing to the spread of fake news. A potential solution to this problem is to maintain a history of what has happened to an image as evidence of the

operations performed, enabling actions that may range from personal to legal measures. This history or "electronic lineage" of the image must be trustworthy; therefore, we propose the use of Blockchain. However, the most widely used public blockchains are slow, consume large amounts of energy, and are inefficient for image storage. In view of this, we propose the use of the Solana blockchain together with the IPFS protocol for image storage. As a result, a distributed, efficient, and reliable scheme can be achieved, which could later be applied to the traceability of various digital assets.

Keywords: Blockchain, raceability, symmetric encryption.

1. Introducción

El control que la gente tiene sobre sus activos digitales una vez que están en internet es debatible, estos activos pueden ser duplicados con facilidad o caer en manos de un tercero que no necesariamente es de confianza. Con esto surge la necesidad de controlar y saber qué sucede con dichos activos digitales.

Siendo las imágenes un ejemplo de activo digital, existen casos en donde imágenes públicas son editadas sin el consentimiento del dueño [16,19] o imágenes compartidas de manera confidencial son compartidas a terceros sin el consentimiento del dueño [22] generando daños personales o siendo parte del fenómeno de *fake news*.

Dada la naturaleza abierta del internet, borrar un activo digital se vuelve sumamente retador. Una solución alternativa es tener la historia de lo que ha sucedido con dicho activo como evidencia de las operaciones realizadas y de ahí tomar acciones que pueden ir desde lo personal hasta lo legal. Esta historia se denomina linaje electrónico. El "linaje electrónico" nos permite conocer el origen y los procesos por los que pasa un activo digital generando la propiedad de trazabilidad [14]. Este registro de procesos da la posibilidad al dueño del activo de saber qué es lo que ocurre con su propiedad. Para lograr esto, las aplicaciones deben generar documentación de los procesos que le están ocurriendo a los activos digitales y almacenarlos de forma segura para que no sean alterados y sean confiables.

La primeras propuestas para crear y almacenar el linaje electrónico involucran un sistema centralizado donde se encuentran todos los registros de procesos [14]. Esta situación obliga a tener un tercero de confianza sin garantía alguna de su comportamiento. Para resolver esta problemática se propone el uso esquemas distribuidos y descentralizados, una opción es Blockchain. Blockchain es una base de datos distribuida cuya tecnología engloba el uso de criptografía, algoritmos de consenso y modelos económicos. Blockchain combina redes punto a punto y algoritmos de consenso distribuidos para resolver problemas de sincronización que aparecen en bases de datos distribuidas tradicionales [9]. Sin embargo, las Blockchain públicas más usadas son lentas y consumen altas cantidades de electricidad, además de no ser eficientes en el almacenamiento de archivos [9]. Esto es consecuencia del tipo de protocolo de consenso usado.

Existen propuestas de almacenar el linaje electrónico usando Blockchain. Azaria et al. [5] proponen un esquema descentralizado para el manejo de registros médicos con el uso de Blockchain. Por otro lado, Sifah et al. [17], presentan el uso de Blockchain para el linaje electrónico archivos en la nube. Finalmente, Khatal et al. [10] proponen de igual forma un marco de referencia para el linaje electrónico de archivos con el uso de Blockchain y de IPFS.

La ventaja de estas propuestas es que ofrecen integridad del linaje electrónico gracias al uso de Blockchain pero tienen como desventaja el hecho de que el almacenamiento de los archivos es centralizado ya que el primero pretende que la información médica se mantenga en las bases de datos de los proveedores mientras que el segundo trabajo es específicamente diseñado para información almacenada en la nube, lo cual significa que un proveedor de servicio de la nube se encarga de administrar los archivos. La tercer propuesta sí maneja descentralización para el almacenamiento de los archivos pero únicamente se concentra en archivos de texto, mientras que este trabajo busca proporcionar una solución para imágenes.

Otro aspecto que vale la pena mencionar es que todas las propuestas anteriores utilizan la red de Ethereum, haciendo de estas soluciones lentas y costosas. El uso de otras redes que implementan protocolos de consenso más eficientes y con un menor costo en las transacciones genera una solución más factible de implementar.

En vista de esto se propone el uso de la Blockchain llamada Solana, basada en un protocolo de consenso eficiente y sustentable que se describe en el trabajo de Yakovenko [21]. Esta Blockchain almacenará el linaje electrónico de imágenes, es decir, las operaciones que se realizarán sobre ellas. Para el caso del almacenamiento de imágenes, se propone el uso de otro protocolo diseñado para este fin llamado IPFS [6].

Como consecuencia se obtendrá una solución distribuida, eficiente y confiable debido a las propiedades ofrecidas por estas tecnologías. Esto se formalizara en un marco de referencia que permitirá verificar su correcto funcionamiento, además de su uso en otros activos digitales que no sean necesariamente imágenes.

2. Linaje electrónico y blockchain

La palabra “linaje” puede definirse como la derivación desde un origen particular hasta un estado específico de un elemento. Esta idea puede aplicarse al mundo digital dando origen al linaje electrónico que no es más que el proceso que conduce a un dato [14]. El linaje electrónico apoya la información y la integridad del proceso documentando las entidades, sistemas y procesos que operan y contribuyen a los datos de interés, sirviendo como un registro histórico inalterable de la duración de los datos y sus orígenes [12]. Sus usos son diversos e incluyen el evaluar la calidad de la información [2], atribuir el origen de un resultado computacional, reproducir ejecuciones previas de aplicaciones o como evidencia en auditorías electrónicas como se menciona en [3]. Muchas de estas

aplicaciones se crearon en entornos centralizados asumiendo que existían las medidas necesarias para garantizar la integridad de esta información.

Por otro lado existe Blockchain que es una lista creciente de registros llamados bloques o transacciones, unidos entre sí a través del uso de funciones hash criptográficas. Una función hash se utiliza para mapear información digital de cualquier longitud a datos digitales de tamaño fijo. Los valores devueltos por este tipo de función se denominan digesto, valores resumen, o simplemente valores hash. Este digesto es único para un valor de entrada dado, así cualquier cambio en los datos de entrada cambia de manera significativa el dato de salida [1], siendo este una herramienta para verificar la integridad de datos. Mediante el uso de estas funciones, las redes punto a punto y los protocolos de consenso [7], Blockchain hace que la historia de activos digitales sea inalterable, inmutable y transparente [18]. Esto se puede ver en aplicaciones como criptomonedas, registros de propiedad, instrumentos financieros, servicios de identidad, cadenas de suministros, IoT, sistemas de votación, entre otros [11].

Existen tres tipos de blockchain: públicas, privadas y consorcio. En las blockchain públicas cualquiera en la red puede validar transacciones y formar parte del proceso para lograr consenso y así tener acceso al historial completo de las transacciones dentro de esta blockchain. Las transacciones además de contener la información de dicha transacción también contiene la firma digital de su creador. Esto quiere decir que cada participante deberá tener un par de llaves pública - privada por cada transacción. La llave privada es usada para firmar digitalmente la transacción y la llave pública para identificar al origen de dicha transacción [1]. Por otro lado, en las blockchains privadas, los nodos deben ser autenticados exitosamente para participar en el proceso de verificación y validación de transacciones y tener acceso a la información dentro del blockchain. Como consecuencia, en este tipo de blockchain, los nodos son identificados a través de un proceso de autenticación. Finalmente, una blockchain de tipo consorcio es una combinación de las dos anteriores donde un conjunto de nodos predeterminados que cuentan con la infraestructura necesaria (usualmente parte de una empresa u organización) validan transacciones y mantienen el blockchain, y los usuarios autenticados envían transacciones y consultas de la información contenida en dicha blockchain. Este esquema permite tener además de autenticación, control de acceso sobre la información [9].

Tanto las Blockhains privadas como las de tipo consorcio no son completamente descentralizadas, por lo que para esquemas descentralizados es conveniente hacer uso de blockchains públicas. Ejemplos de estas blockchains son Bitcoin y Ethereum siendo las primeras en ser creadas y de las que se encuentran diversas herramientas y documentación. Sin embargo, tienen varias desventajas como que la verificación de transacciones es lenta y su gasto de energético es alto. Además, cada transacción tiene un costo, usualmente bastante elevado, esto debido a que hacen uso del protocolo de consenso de prueba de trabajo (o Proof of Work) [20].

Para solucionar este problema, han surgido alternativas que buscan ser más eficientes en costo, tiempos de transacción y uso energético. Esto lo logran usando

otros tipos de protocolos de consenso siendo el más popular Proof of Stake (PoS). Una de estas alternativas es la red Solana [21] que propone el uso del protocolo de consenso PoS, donde cada miembro de la red apuesta una cierta cantidad de tokens de la propia red para tener la posibilidad de generar un bloque. Adicionalmente incluye el uso de un segundo protocolo llamado Proof of History (PoH), en el que una secuencia de hashes es usada como un registro del tiempo creando un orden en las transacciones y proporcionando una rápida sincronización entre los nodos de la red [21]. Estos dos, hacen de Solana una red más eficiente que las tradicionales como puede verse en [15].

Un elemento importante para el desarrollo de aplicaciones sobre blockchain son los contratos inteligentes. Un contrato inteligente es un programa de computadora auto-verificable, auto-ejecutable y resistente a la manipulación que consiste en un conjunto de reglas que se ejecutan en la Blockchain. Este programa permite ejecutar código sin la necesidad de un tercero, tomando una transacción como entrada, ejecutando el código correspondiente y desencadenando los eventos de salida [13].

Una limitante de Blockchain, como consecuencia del tiempo que requiere el verificar una transacción, es la imposibilidad de almacenar archivos. La mayoría de las soluciones existentes no contemplan el almacenar archivos completos dentro de un bloque que pertenezca a una blockchain. Con el objetivo de mantener la descentralización y distribución de dichos archivos, se requiere del uso de los llamados "sistemas distribuidos de archivos". Un ejemplo de este tipo es el Sistema de Archivos Andrew o Andrew File System (AFS) [4]. Otros ejemplos son las aplicaciones punto a punto para compartir archivos de gran tamaño entre las que se encuentran Napster, KaZaA y Bit Torrent. Finalmente está IPFS. InterPlanetary File System o IPFS es un sistema distribuido de archivos que busca conectar todos los equipos de cómputo con el mismo sistema de archivos. A diferencia de los ejemplos anteriores, IPFS está diseñado como infraestructura para construir aplicaciones sobre él. IPFS proporciona un modelo de almacenamiento en bloques con dirección a contenido de alto rendimiento con hiperenlaces con dirección de contenido, formando un árbol de Merkle. Además, IPFS combina una tabla hash distribuida, un intercambio de bloques incentivado y un espacio de nombres auto-certificable [6], haciéndolo una buena solución cuando se desea descentralización y distribución de archivos.

Usando las tecnologías mencionadas, se creará un esquema de trazabilidad de imágenes distribuido, eficiente y confiable que posteriormente puede ser usado en dar trazabilidad a otros activos digitales que no necesariamente sean imágenes. A continuación se presenta la arquitectura de este esquema usando la blockchain de Solana e el sistema de archivos IPFS.

3. Arquitectura del esquema de trazabilidad de imágenes basado en blockchain

La arquitectura del esquema propuesto se muestra en la Figura 1. Esta figura muestra la manera en que un usuario se comunica con una interfaz

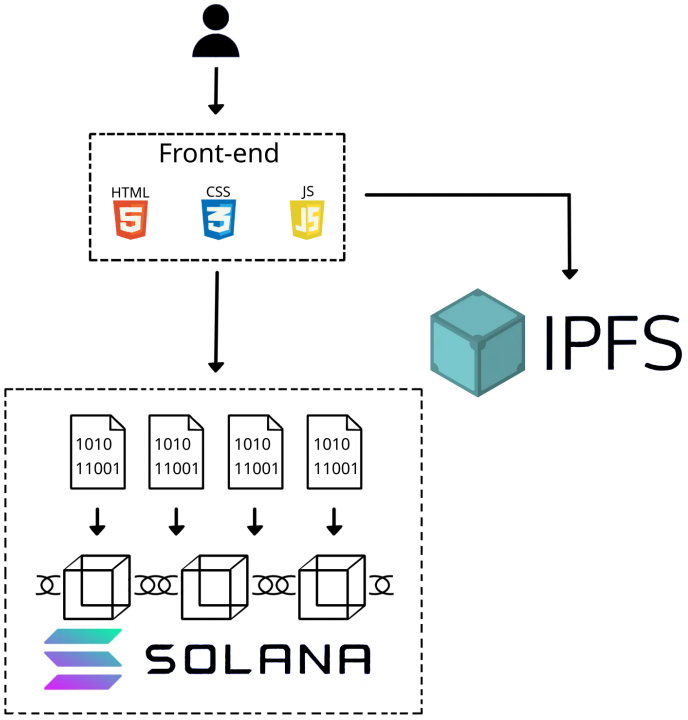


Fig. 1. Arquitectura general.

o *front-end* dentro de un navegador. Desde ahí, el usuario podrá solicitar el realizar funciones sobre las imágenes que desea compartir o descargar. A través de contratos inteligentes (mostrados en la figura como pequeños cuadros) que son ejecutados en la Blockchain se realizan las operaciones sobre las imágenes y a su vez se registran estas operaciones como transacciones en la misma Blockchain. Por último, la o las imágenes que sean parte de las operaciones realizadas son almacenadas en IPFS, desde donde podrán ser accedidas en cualquier momento.

Esta figura también muestra que nuestra propuesta hace uso de las tecnologías descritas en la Sección 2. A continuación se presentan los resultados de analizar que funciones sobre imágenes el usuario realizará.

3.1. Casos de uso

A partir de un análisis de lo que un usuario desea realizar con sus imágenes se definieron las operaciones que nuestro esquema debe soportar. Con esto, se determinó que información será almacenada dentro de la blockchain así como el diseño de los correspondientes contratos inteligentes. Así, se definieron las siguientes operaciones a realizar sobre las imágenes en nuestro esquema.

1. **Subir imagen.** Un usuario desea subir una imagen a la blockchain para compartirla con todos los usuarios de esta. Esta imagen puede subirse en claro, para que cualquier usuario pueda verla, o cifrada, para que sólo los usuarios con las llaves correspondientes pueden ver la imagen original.
2. **Modificar permisos de una imagen.** Cada imagen se subirá con ciertos permisos (público o editable). Estos permisos pueden ser modificados durante el ciclo de vida de la imagen.
3. **Descargar imagen.** Una imagen puede ser descargada de manera local si se tienen los permisos.
4. **Editar imagen.** Una imagen puede ser editada de manera local si se tienen los permisos.

A continuación se presenta una descripción a detalle del funcionamiento de estos 4 casos de uso.

Subir imagen La operación de subir una imagen, mostrada en la Figura 2, involucra un contrato inteligente que permitirá que el usuario agregue una imagen a IPFS y la evidencia de esta acción en la red Blockchain de Solana. Lo que hará dicho contrato inteligente es guardar una estructura de datos en la blockchain, la cual contiene la siguiente información de la imagen.

1. La llave púbica del usuario que sube la imagen.
2. El identificador de contenido (`cid`) de la imagen en donde `cid=hash(imagen)`. Este es un identificador único se utiliza IPFS para identificar la imagen de manera única.
3. El `cid` de la imagen padre de la cual proviene la imagen a subir. Esto aplica cuando la imagen a subir sea resultado de una edición.
4. Un indicador booleano para saber si la imagen es una edición.
5. Un indicador booleano para identificar si la imagen es la diferencia entre la imagen original y la edición.
6. Un indicador booleano para saber si la imagen es pública.
7. Un indicador booleano para establecer si la imagen es editable.

Al subir la imagen, el primer paso consiste en ejecutar el correspondiente contrato inteligente para almacenar la estructura previamente descrita dentro de la blockchain (paso 1). Una vez almacenada, habrá una respuesta exitosa por parte de la blockchain (paso 2) y entonces la imagen podrá subirse a IPFS donde permanecerá disponible, completando así el paso 3. De esta manera la imagen estará almacenada en IPFS y en la blockchain de Solana sólo se guardará la referencia única a dicha imagen.

Algo a tomar en cuenta es que el contenido en IPFS es público, esto quiere decir que cualquiera que tenga el correspondiente `cid` de la imagen podría acceder ella realizando una consulta. Por esta razón es necesario brindar la opción de cifrado para que el usuario tenga mayor control de su imagen. En caso de que el usuario quiera que su imagen sea privada, esta imagen se cifrará antes de ser subida a IPFS, almacenando en la blockchain el identificador de contenido de la

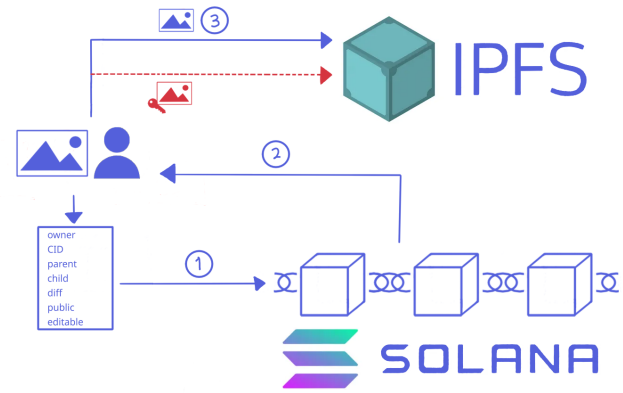


Fig. 2. Proceso de subir una imagen.

imagen ya cifrada. Este paso opcional puede observase en la figura 2 dentro del paso 3 en color rojo.

El usuario puede elegir al momento de subir una imagen si ésta es pública o privada y si es editable o no. Si la imagen es privada, será cifrada para que solo las personas permitidas tengan acceso a ésta. En ese caso, el dueño de la imagen creará una llave de cifrado simétrico con la que dicha imagen se cifrará.

La imagen 3 muestra este proceso, en donde el usuario que solicite la llave subirá a la blockchain una estructura de datos que contenga la llave pública de dicho usuario, una segunda llave pública RSA con la que se cifrará la llave simétrica, el `cid` de la imagen a la que desea acceder y un espacio en blanco donde la llave para descifrar la imagen será guardada. Posteriormente el dueño, de aceptar compartir la llave, subirá la llave simétrica cifrada con la llave pública RSA en el espacio vacío antes mencionado, el solicitante obtiene la llave y por último se elimina la información de la blockchain. De esta manera se comparte la llave de cifrado de manera segura sólo a la persona que tendrá permiso para descifrar esta imagen.

Modificar permisos Como se mencionó anteriormente, al momento de subir una imagen el usuario puede decidir si ésta es pública o privada y si es editable o no. A esto le llamamos permisos de imagen. Para modificar estos permisos se utilizará un contrato inteligente que, dependiendo de la elección del usuario, asignará uno de los permisos a la información de la imagen que se encuentra en blockchain asociada al `cid` de la imagen. Esto se presenta la figura 4. Los tipos de permisos son los siguientes.

– `Public.` Este es un campo booleano que cuando esta en 1 indica que el archivo será público, en este caso cualquier persona puede acceder a la imagen sin ninguna restricción. Si quisiéramos compartir la imagen con algún usuario

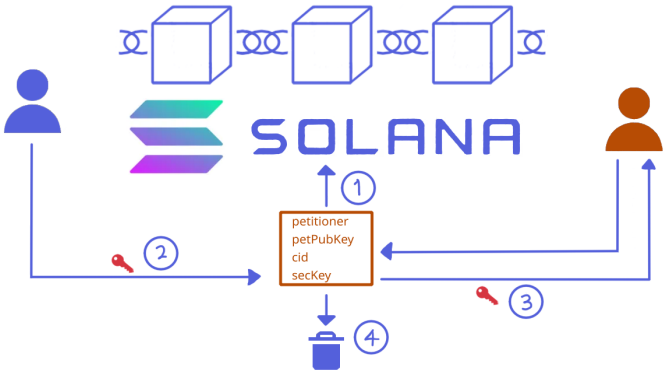


Fig. 3. Proceso de compartir llave.

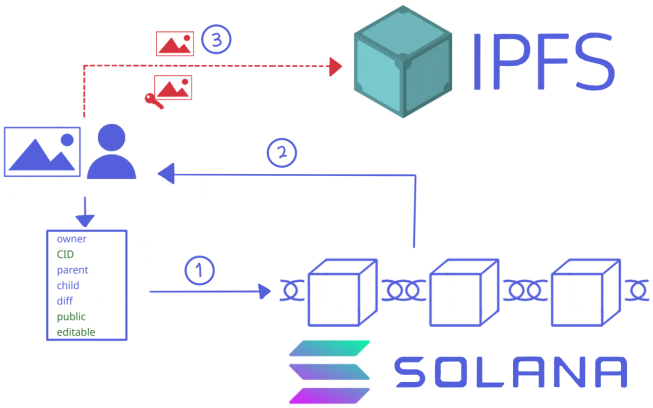


Fig. 4. Proceso de modificar permisos de una imagen.

en específico, el cual podrá consultar el archivo almacenado en IPFS, este campo estaría en 0. En esta caso la imagen estaría cifrada y se tendría que solicitar la llave simétrica correspondiente de descifrado.

– `Editable`. Este es un campo booleano que cuando esta en 1 indica que la imagen puede ser descargada directamente de IPFS para posteriormente ser editada. Un usuario con esta autorización, podrá descargar la imagen correspondiente y se generará un registro en blockchain de que esta imagen fue obtenida con el fin de realizar una modificación. Si este campo esta en 0 la imagen no podrá descargarse y como consecuencia no será editable, es decir, será de sólo lectura.

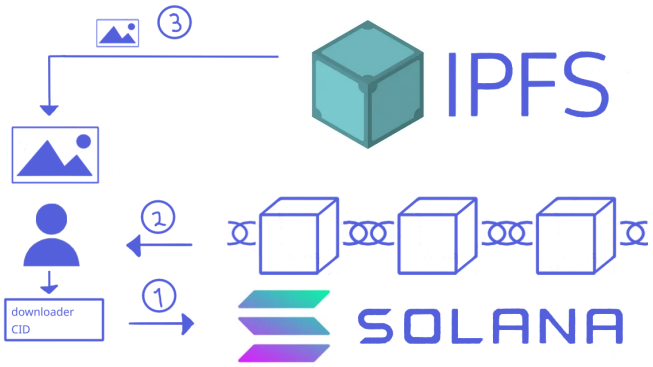


Fig. 5. Proceso de descargar una imagen.

Un usuario tiene la opción de modificar los permisos de una imagen que subió previamente. De ser así, los campos booleanos de público y editable así como el campo que contiene el `cid` de la imagen pueden ser modificados. Para modificar los permisos, el contrato inteligente actualizará los campos antes mencionados (resaltados con color verde en la figura 4) con los nuevos valores, guardando los cambios en la blockchain en el paso 1. Si el usuario decide cambiar su imagen de pública a privada o viceversa (paso 2), está deberá cifrarse o descifrarse dependiendo del cambio y por lo tanto tendrá que volverse a subir el nuevo archivo a IPFS y modificarse el `cid` que está guardado en la blockchain (paso 3).

Descargar imagen Si el usuario desea descargar una imagen, se crea una estructura de datos que incluye la llave pública del usuario que descarga dicha imagen y el `cid` de la imagen que se descarga. La figura 5 muestra este proceso. En el paso 1 se guarda un bloque de esta acción, esto es, indicando que un usuario descargó una imagen. Este bloque se guarda en la Blockchain. Si el usuario que solicita descarga no tiene permiso de edición no podrá realizar dicha descarga y requerirá que el dueño de la misma cambie los permisos. En caso de que sí exista el permiso (paso 2), la imagen puede ser descargada sin importar si es o no pública con la única consideración de que se deberá solicitar la llave correspondiente para descifrar la imagen en caso de que esta sea privada (paso 3).

Editar imagen La edición de una imagen es una combinación de las operaciones de descargar y subir imagen como puede observarse en la figura 6. Primero se ejecuta el contrato inteligente para descargar la imagen (Pasos del 1 al 4), posteriormente se ejecuta otro contrato inteligente para subir la imagen editada (Pasos del 5 al 6) y, por último, un tercer contrato inteligente es ejecutado para para subir otra imagen que muestra las diferencias entre la imagen original y la modificada (Pasos del 7 al 9). Ambas imágenes se suben a IPFS para alimentar el linaje electrónico de la imagen que las originó, mostrando los cambios por los que

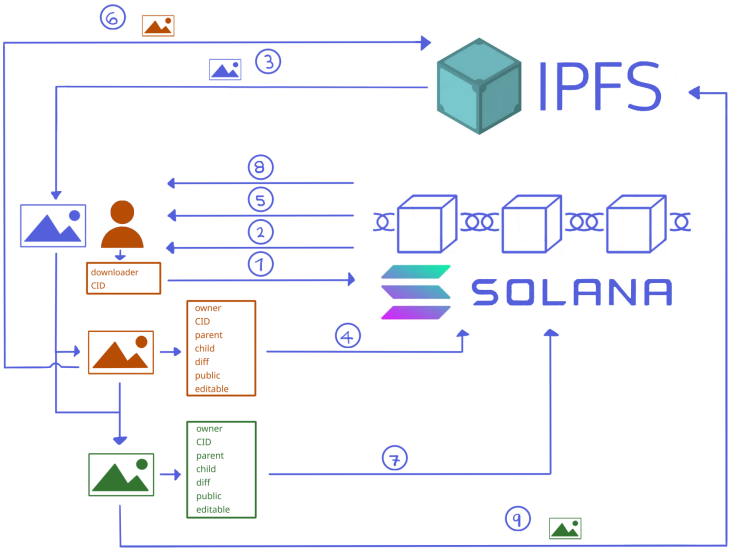


Fig. 6. Proceso de edición de una imagen.

pasó la misma. De esta manera generamos evidencia de cualquier modificación realizada a las imágenes dentro de nuestro sistema.

Profundizando un poco más en la comparación de imágenes se propone utilizar la biblioteca `jimp` (JavaScript Image Manipulation Program) de Javascript [8]. Esta implementa el método `diff` que obtiene las diferencias entre imágenes generando una nueva imagen que únicamente muestra las diferencias entre ambas imágenes. En la figura 7 puede observarse un ejemplo del proceso de comparación. Se muestra la imagen original, la imagen editada (en esta caso se cambió el color del fondo) y a la derecha la imagen resultante del proceso de comparación que muestra la modificación realizada. Estas tres imágenes son guardadas en blockchain mostrando cómo la nueva imagen editada es el resultado de la diferencia y la imagen original.

Para utilizar este método se deben tomar en consideración varios aspectos. El formato de las imágenes debe ser uno de preservación como el formato PNG. Otro aspecto a considerar es que las imágenes deben ser del mismo tamaño ya que la comparación generaría un empalme o diferencia en el resultado que no permitirá ver las modificaciones de forma clara.

Usando este análisis es posible realizar la implementación de los casos de uso descritos en forma de contratos inteligentes. En la siguiente sección se presenta dicha implementación.

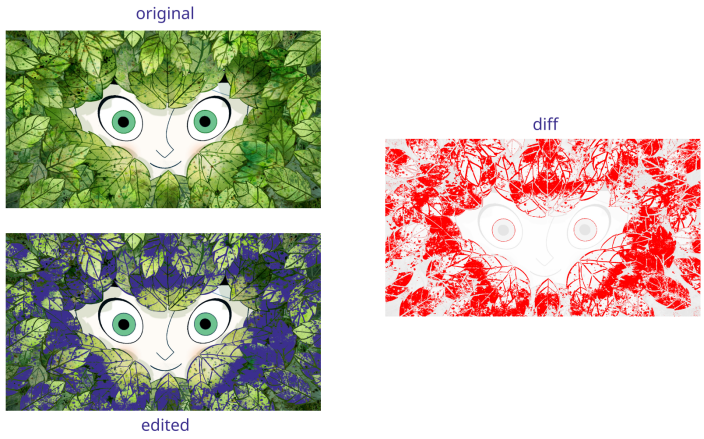


Fig. 7. Comparación de la imagen original y la modificada.

```
struct ImgData {
 owner: String,
 CID: String,
 parent: String,
 child: bool,
 diff: bool,
 public: bool,
 editable: bool
}
```

```
struct DwnldLog {
 downloader: Pubkey,
 CID: String
}
```

```
struct SecKeyRequest {
 petitioner: String,
 petitionerPublicKey: String,
 CID: String
 secretKey: Array,
}
```

Fig. 8. Estructuras de datos que se guardarán en la Blockchain.

4. Resultados

La implementación de los correspondientes contratos inteligentes se realizó utilizando los lenguajes de programación Rust y Typescript, siendo el primero necesario para los contratos inteligentes y el segundo para la interacción del usuario con la Blockchain y con IPFS. Para almacenar la información de una operación relacionada con una imagen se requiere crear una cuenta que incluye una llave privada y una pública. Además esa cuenta requiere de cierta cantidad de SOL (la criptomoneda con la que se pueden realizar operaciones en la Blockchain de Solana). En dicha cuenta se guardarán las estructuras mostradas en la Figura 8 ya descritas en la sección 3.1. Estas estructuras cambian de elementos dependiendo el caso de uso a ejecutarse. Además, en esta cuenta se almacenan y ejecutan los contratos inteligentes que implementan estos casos de uso. En esta sección nos enfocaremos en estos contratos.

Cada contrato inteligente en Solana consiste de los siguiente 6 archivos escritos en lenguaje de programación Rust.

1. `entrypoint.rs` Este es la entrada al programa y donde se enviará la información que se desea guardar en la Blockchain.

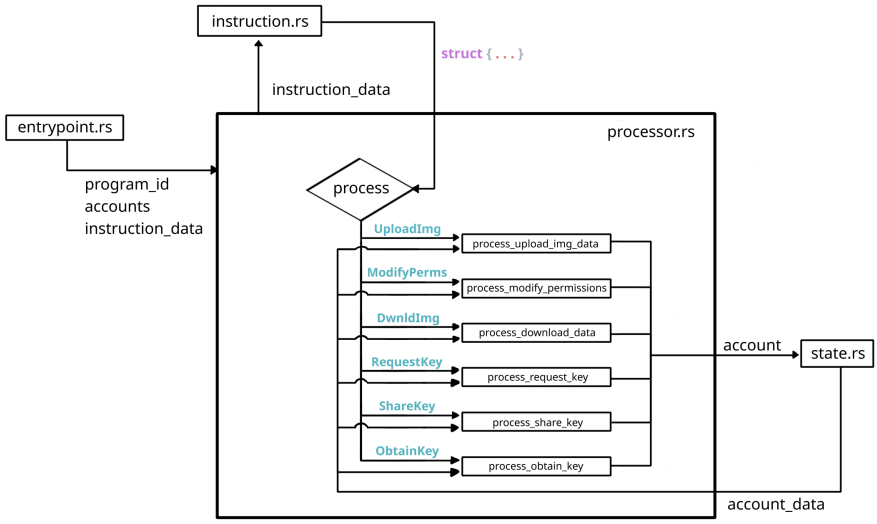


Fig. 9. Diagrama de flujo de la ejecución de un contrato inteligente.

2. `instruction.rs` En este se elige qué operación se va a realizar sobre la imagen de las descritas en la Sección 3.
3. `processor.rs` Aquí se ejecuta la operación seleccionada en `instruction.rs`.
4. `state.rs` Se define cómo se va a guardar la información en la Blockchain.

La figura 9 muestra un diagrama de flujo que describe la interacción entre los archivos de Rust. El archivo `instruction.rs` contiene las diferentes opciones que se pueden ejecutar en `processor.rs`, esto dependiendo del tipo de información de entrada en `entrypoint.rs`. Esta información es un arreglo de bytes donde el primer byte corresponde al número de instrucción (`process`) que se va a ejecutar y el resto es la información que se desea ingresar a la Blockchain. Existen 6 diferentes operaciones. Una vez elegida alguna de estas se realiza el llamado a la función correspondiente. En la imagen podemos ver que estas funciones tienen su equivalente a los Casos de Uso descritos previamente.

Por ejemplo, si se elige la opción 0 que corresponde a subir una imagen, la información dentro del arreglo de bytes que se ingresó al programa pasará a `processor.rs` donde se ejecuta la función $process\_upload\_img\_data()$. En esta función se agrega la información a la Blockchain dentro de una cuenta creada para ese fin. Esta información es la de la estructura mostrada en la figura 10. La figura 11 muestra el registro de dicha transacción en el explorador de Solana. Una vez que la transacción es aprobada en Blockchain es cuando la imagen se sube a IPFS. De esta manera garantizamos que sólo existan imágenes de transacciones validadas.

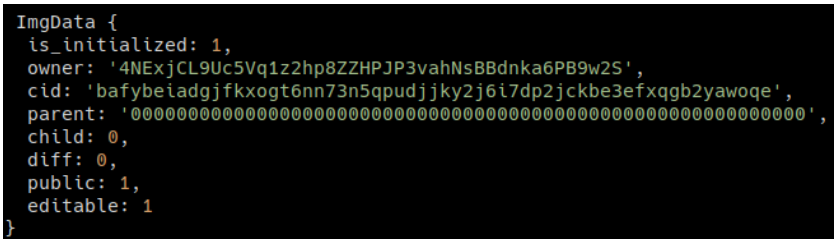


Fig. 10. Información de la imagen disponible en la Blockchain.

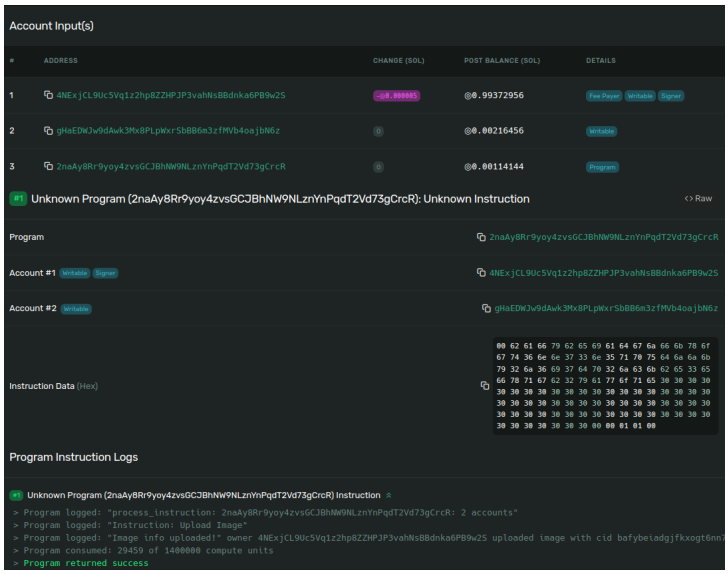


Fig. 11. Explorador de la Blockchain Solana.

Si se accede a la imagen desde IPFS mediante el identificador de contenido almacenado en la Blockchain (figura 12), se puede corroborar que el proceso tuvo éxito y efectivamente la imagen esta en IPFS.

4.1. Consulta del linaje electrónico de una imagen

Una vez que las operaciones realizadas sobre las imágenes han sido documentadas y almacenadas en la Blockchain de Solana e IPFS, se pueden realizar consultas que muestren el linaje electrónico de estas imágenes.

Estas consultas son realizadas haciendo uso de los mensajes de registro del contrato inteligente emitidos en cada transacción. Durante la ejecución del contrato inteligente se generan ciertos mensajes que indican qué imagen está involucrada en la operación (identificada por el `cid`), de qué operación se trata (de las mostradas en la figura 9) y el usuario que está ejecutando la operación

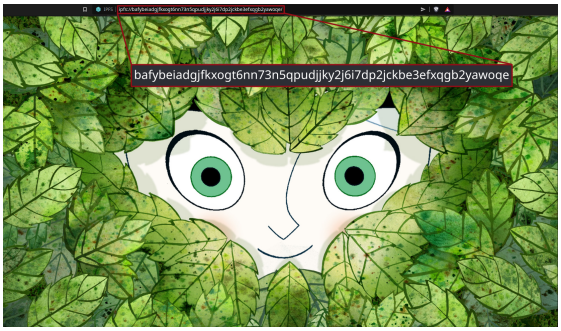


Fig. 12. Imagen vista desde IPFS.

Fig. 13. Imagen y su respectiva información.

(identificado por su llave pública), además de información adicional que depende de la misma operación.

De esta manera se realiza una búsqueda de las transacciones realizadas por un usuario donde el `cid` involucrado coincida con el `cid` de la imagen de interés. Al realizar esto, será posible observar los procesos que tuvieron lugar e ir construyendo el linaje de la imagen través de los registros en las transacciones dentro del Blockchain.

La figura 13 muestra una imagen almacenada en IPFS y su respectiva información almacenada en la Blockchain de Solana. Sabemos que dicha imagen es una edición de otra imagen, ya que su información muestra el indicador `child` con un valor de 1 indicando la presencia de una imagen padre.

A partir del `cid` de esta imagen padre se inicia la consulta del linaje electrónico (figura 14), observando la transacción que registró la edición de la imagen se tiene el `cid` (en color verde) que corresponde a la imagen mostrada en la figura 13 y el `cid` de la imagen padre (en color rojo).

Al consultar las transacciones de la imagen padre se obtienen 4 transacciones (Tx_i) mostradas en la figura 14. En esta imagen podemos observar que sólo la Tx_3 corresponde a la edición de una imagen, dicha imagen (con `cid` en color morado) resulta ser la imagen original. Al consultar el resto de las transacciones se observa un proceso de edición, un cambio de permisos y la subida de la imagen padre.

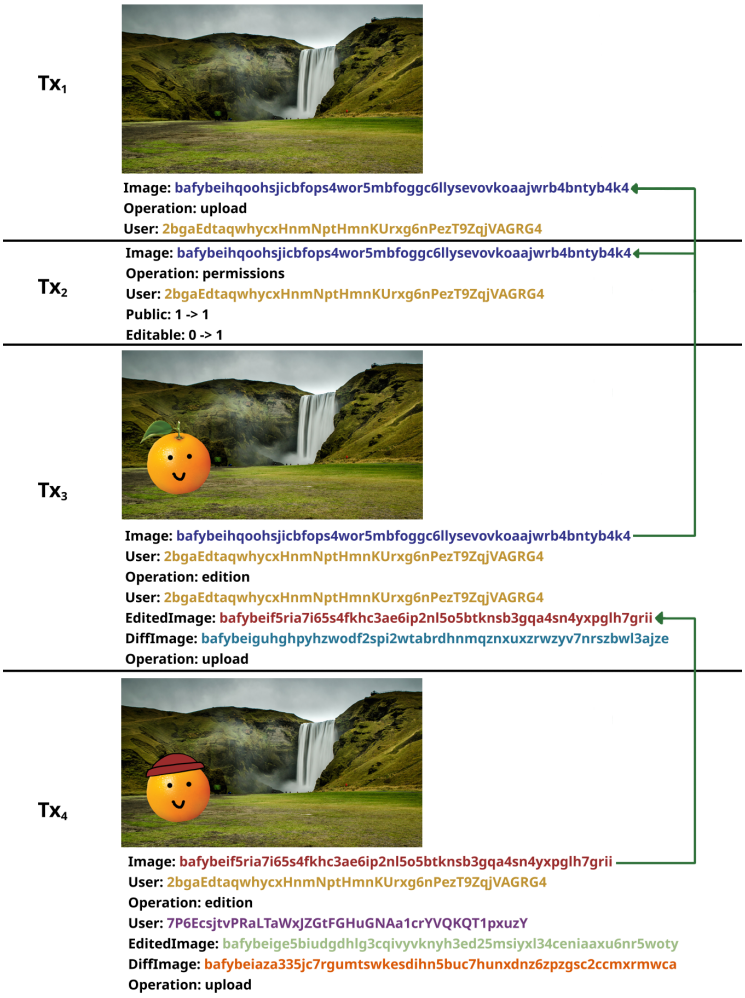


Fig. 14. Muestra de la consulta del linaje electrónico de la imagen.

Así, el linaje completo de este ejemplo puede describirse de la siguiente manera. La imagen con `cid` en color morado fue subida a IPFS siendo pública y de solo lectura (Tx_1), posteriormente el usuario decidió modificar uno de los permisos y hacer la imagen en editable (Tx_2). En algún momento el mismo usuario realizó una edición y la documentó (Tx_3). Finalmente, esta imagen resultado fue nuevamente editada pero ahora por el usuario con llave pública en color morado (Tx_4).

Todo esto se realiza utilizando los mensajes creados específicamente para documentar dentro de los registros de las transacciones las operaciones que se realizan, facilitando así la consulta del linaje electrónico.

5. Conclusiones

Es clara la necesidad que existe de que las personas tengan control sobre la información que comparten en Internet. La utilización de Blockchain para almacenar el linaje electrónico de imágenes brinda integridad y no repudio a la información referente a la historia de dicho activo. Sin embargo, esto requiere de una tecnología eficiente, sustentable y que permita el almacenamiento de archivos.

En vista de esto, el trabajo presentado incluye el uso del protocolo de consenso PoS con PoH y el protocolo de almacenamiento IPFS. Esto nos permitirá documentar los cambios que sufren las imágenes usando contratos inteligentes con tiempos de transacción cortos y almacenamiento distribuido. También, el construir este esquema sobre una arquitectura distribuida permite confiar en toda la información generada, garantizando que las medidas de control y trazabilidad implementadas serán cumplidas.

Se logró realizar una implementación del esquema propuesto haciendo uso de la Blockchain de Solana a través de los lenguajes de programación Rust y Typescript. Esta implementación logra registrar y ejecutar las operaciones de subir una imagen, modificar sus permisos, descargarla y editarla, además de proporcionar métodos para generar y compartir llaves para cifrar y descifrar simétricamente imágenes que son clasificadas como privadas. Finalmente, se implementó la consulta del linaje electrónico y su posterior análisis.

Como parte del trabajo futuro está el realizar una implementación completa que incluya el desarrollo de un *front-end* para que usuarios finales puedan ser parte de este sistema. A la vez, teniendo mucha más información relacionada al uso de imágenes, se podría desarrollar un contrato inteligente que permita la visualización de las consultas de la historia de las imágenes de forma más amigable para el usuario, de tal forma que cada dueño de una imagen tenga información de cómo están siendo usadas sus imágenes.

Acknowledgements. Se agradece a DGAPA por el proyecto PAPIIT TA101021 y a CONACyT por el apoyo a través de la beca de maestría 1085102.

Referencias

1. Aldeco-Pérez, R., Aguilar Torres, G., Cruz Cortés, N., Domínguez Perez, L. J., Escamilla Ambrosio, P. J., Gallegos García, G., León Chavez, M. A., Monroy Borja, R., Rodríguez Henríquez, L. M., Rodríguez Henríquez, F. J., Rodríguez Mota, A., Salinas Rosales, M., Silva Trujillo, A. G.: Introducción a la Ciberseguridad y sus aplicaciones en México. Academia Mexicana de Computación, A. C., 1 edn. (2020), http://amexcomp.mx/files/LibroCiber-ISBN-V2.pdf
2. Aldeco-Pérez, R., Leon Chavez, M.: Evaluar la Calidad de los Objetos de Aprendizaje Mediante Linaje Electrónico. In: El Desarrollo de los Recursos Digitales para la Educación en México, pp. 240. Benemérita Universidad Autónoma de Puebla, 1st edn. (2013), https://tinyurl.com/2p8akmch

3. Aldeco-Pérez, R., Moreau, L.: Securing provenance-based audits, vol. 6378 LNCS. Springer (2010)
4. Arpaci-Dusseau, R. H., Arpaci-Dusseau, A. C.: The Andrew File System (AFS). In: Arpaci-Dusseau Books, pp. 1—-14 (2014), https://pages.cs.wisc.edu/∼remzi/OSTEP/dist-afs.pdf
5. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: Medrec: Using blockchain for medical data access and permission management. In: 2016 2nd International Conference on Open and Big Data (OBD). pp. 25–30 (2016) doi: 10.1109/OBD.2016.11
6. Benet, J.: Ipfs - content addressed, versioned, p2p file system (2014)
7. González-Ortega, A., de Asís López-Fuentes, F.: A web-based didactic tool for teaching of distributed consensus. Res. Comput. Sci., vol. 148, no. 5, pp. 25–32 (2019)
8. JIMP: `www.npmjs.com/package/jimp`
9. Joshi, A., Han, M., Wang, Y.: A survey on security and privacy issues of blockchain technology. Mathematical Foundations of Computing, vol. 1, pp. 121–147 (01 2018) doi: 10.3934/mfc.2018007
10. Khatal, S., Rane, J., Patel, D., Patel, P., Busnel, Y.: Fileshare: A blockchain and ipfs framework for secure file sharing and data provenance. In: Patnaik, S., Yang, X.-S., Sethi, I. K. (eds) Advances in Machine Learning and Computational Intelligence. pp. 825–833. Springer Singapore, Singapore (2021)
11. Mattila, J.: The blockchain phenomenon – the disruptive potential of distributed consensus architectures. ETLA Working Papers 38, Helsinki (2016), http://hdl.handle.net/10419/201253
12. McDaniel, P.: Data provenance and security. IEEE Security Privacy, vol. 9, no. 2, pp. 83–85 (2011) doi: 10.1109/MSP.2011.27
13. Mohanta, B. K., Panda, S. S., Jena, D.: An overview of smart contract and use cases in blockchain technology. In: 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). pp. 1–4 (2018) doi: 10.1109/ICCCNT.2018.8494045
14. Moreau, L., Groth, P., Miles, S., Vázquez-Salceda, J., Ibbotson, J., Sheng, J., Munroe, S., Rana, O., Schreiber, A., Tan, V., Varga, L.: The provenance of electronic data. Commun. ACM, vol. 51, pp. 52–58 (04 2008) doi: 10.1145/1330311.1330323
15. Pierro, G. A., Tonelli, R.: Can solana be the solution to the blockchain scalability problem? In: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). pp. 1219–1226 (2022) doi: 10.1109/SANER53432.2022.00144
16. Pixel, M.: México sigue teniendo problemas con las fake news: algunas de estas fotos son reales pero no son de Cancún (2017), https://www.xataka.com.mx/otros-1/mexico-sigue- teniendo-problemas-con-las-fake-news-algunas-de-estas-fotos-son-reales-pero-no-son-de-cancun
17. Sifah, E. B., Xia, Q., Agyekum, K. O.-B. O., Xia, H., Smahi, A., Gao, J.: A blockchain approach to ensuring provenance to outsourced cloud data in a sharing ecosystem. IEEE Systems Journal, pp. 1–12 (2021) doi: 10.1109/JSYST.2021.3068224
18. Sivleen, K., Sheetal, C., Aabha, S., Jayaprakash, K.: A research survey on applications of consensus protocols in blockchain. Security and Communication Networks, vol. 2021 (01 2021) doi: 10.1155/2021/6693731

19. Wen, T.: The hidden signs that can reveal a fake photo (2020), https://www.bbc.com/future/article/20170629-the-hidden-signs-that-can-reveal-if-a-photo-is-fake
20. Xiao, Y., Zhang, N., Lou, W., Hou, Y. T.: A Survey of Distributed Consensus Protocols for Blockchain Networks. IEEE Communications Surveys and Tutorials, vol. 22, no. 2, pp. 1432–1465 (apr 2019) doi: 10.1109/COMST.2020.2969706
21. Yakovenko, A.: Solana: A new architecture for a high performance blockchain v0.8.13. Tech. rep. (2020)
22. Yañez, B., Galván, M., Ramírez, S.: El ABC de la 'Ley Olimpia': sus alcances y retos (2022), https://politica.expansion.mx/sociedad/2022/21/25/el-abc-de-la-ley-olimpia-sus-alcances-y-retos

ISSN 1870-4069

Motor matemático OpenSource con inyección de dependencias dinámicas en Java

Edgar Hamlet Solano-Díaz, Francisco López-Orozco,
Rogelio Florencia-Juárez, Jesús Israel Hernández-Hernández

Instituto de Ingeniería y Tecnología,
México

Universidad Autónoma de Ciudad Juárez,
México

al154007@alumnos.uacj.mx,
{francisco.orozco,rogelio.florencia,israel.hernandez}@uacj.mx

Resumen. Este artículo presenta la implementación de una herramienta *OpenSource* tipo calculadora científica modular con el objetivo de facilitar el desarrollo de operaciones y funciones matemáticas en una misma aplicación con módulos altamente modificables, completamente personalizables y fáciles de compartir. El programa tiene su centro de ejecución en un motor matemático de operaciones dinámicas donde el usuario evita la necesidad de programar en lenguajes matemáticos especializados o desarrollar toda la lógica matemática de cero en un lenguaje conocido para el uso de una única función. El programa trae consigo todo lo necesario para la generación auto-descriptiva en lenguaje humano de cada operación, el proceso matemático correcto de realizar la jerarquía de operaciones y una ventana de visualización útil para interpretar resultados.

Palabras clave: Motor matemático, inyección de dependencias, programación modular, explicaciones matemáticas auto-generadas.

Open-Source Mathematical Engine with Dynamic Dependency Injection in Java

Abstract. This article presents the implementation of an open-source, modular scientific calculator tool designed to facilitate the development of mathematical operations and functions within a single application. The tool is built with highly modifiable, fully customizable, and easily shareable modules. Its core execution relies on a mathematical engine of dynamic operations, allowing users to avoid programming in specialized mathematical languages or developing the entire mathematical logic from scratch in a known programming language just to use a single function. The program includes everything required for self-descriptive human-readable generation of each operation, ensures the correct

application of the order of operations, and provides a visualization window that helps users interpret results effectively.

Keywords: Mathematical engine, dependency injection, modular programming, auto-generated mathematical explanations.

1. Introducción

La computación aplicada a la resolución de matemática avanzada es un tema recurrente en la programación, este campo ha dado paso a herramientas altamente especializadas en la resolución de ecuaciones complejas y diseños abstractos. El desarrollo de estos programas ha convergido en la creación de diversos software auto-explicativos donde por cada operación el resultado es interpretado y explicado paso a paso la justificación de su operación o han surgido en la creación de nuevos lenguajes de programación altamente especializados en el desarrollo y visualización de abstracciones matemáticas especificas.

El proceso de uso y creación de este tipo de herramientas informáticas aplicadas a la matemática puede verse desde dos perspectivas.

El usuario final, estudiante o docente, que seguido ve en la problemática de las pocas herramientas didácticas disponibles en el mercado, con poca capacidad de personalización, recursos limitados para su uso o pago obligatorio para el completo acceso y poca facilidad para compartir o duplicar. El programador o investigador que desarrolló la herramienta final el cuál pasó por el proceso de capacitación en algún lenguaje matemático altamente especializado o el proceso de creación desde cero de todo un entorno matemáticamente correcto para el desarrollo de las funciones matemáticas a usar.

Estas posiciones se convierten en un ciclo auto-alimentado donde el elevado tiempo de desarrollo por la capacitación en materia de programación y matemática deja de lado la gran comunidad de programadores no especializados en temas de matemáticas o matemáticos no completamente familiarizados con la programación de alto nivel, esto haciendo las posibilidades de nuevas herramientas interactivas muy cerradas subiendo así los costos de uso y la continua sobre especialización del software.

2. Trabajos previos

Son conocidos programas de resolución de problemas paso a paso con explicaciones verbales[2] , programas de representación gráfica matemática[3], lenguajes de programación especializados de forma nativa[1] o algoritmos matemáticos de antemano optimizados para la resolución de ecuaciones jerárquicamente correctas [10].

2.1. Programas de resolución de problemas paso a paso:

Programa que en tiempo real describe por pasos y da justificación a sus operaciones con descripciones auto-generadas y gráficos explicativos.

Actualmente son populares ejemplos de aplicaciones web de este tipo que ofrecen la resolución paso a paso de forma parcial con detalles a cambio de una suscripción paga; igualmente no se encuentra disponible y popularizada un servicio o aplicación donde uno mismo como usuario pueda personalizar las funciones que usará con explicaciones paso a paso.

2.2. Programas de representación gráfica:

Programas de visualización con datos de entrada en diferentes unidades matemáticas; planos cartesianos, mapas 3D, tablas o diversos sistemas de coordenadas. En los ejemplos más populares de este tipo de programas se encuentra la complicación de no tener una forma de representar los datos personalizada o estandarizada ya que cada programa usa el formato que pueda de acuerdo a sus necesidades y no las del usuario.

2.3. Lenguajes de programación especializados en matemáticas:

Lenguajes de programación diseñados con el entorno matemático especializado. Las operaciones necesarias para su procesamiento, graficación e interpretación son funciones nativas del lenguaje. La problemática con este tipo de lenguajes radica en su nivel de especialización y toda la capacitación previa que debe tenerse elevando los costes de desarrollo final.

Finalmente se llega a la conclusión que no se ha encontrado un software popularizado con la capacidad de resoluciones matemáticas paso a paso con entradas variables y adaptables a los requerimientos del usuario, 100 % con salida de datos e interpretación matemática disponible y programada por la misma comunidad, teniendo una oportunidad de desarrollo y popularización entre comunidades de programadores y usuarios no programadores la idea desarrollada en este documento.

3. Metodología

El objetivo principal fue la creación de un motor matemático compatible con números enteros, fracciones, decimales, vectores, ángulos, entre otros; que permitiera fácilmente su implementación para realizar operaciones entre ellos. Además, que esto fuera posible de manera dinámica sin la necesidad de desarrollar múltiples motores para diferentes unidades u operaciones. En conjunto que se tuviera un apartado visual para representar cada tipo de unidad creada e inyectada. Todo complementado con un apartado para la explicación paso a paso de cada operación realizada de forma auto generada, con plantillas descritas por el programador del modulo a usar, y así hacer personalizada la experiencia de crear *"Cuadernos de Apuntes"* como un *"Todo en uno"*. Por ultimo el objetivo se buscaba la facilidad de incluir nuevos módulos útiles para la personalización de los apuntes del usuario como una pizarra virtual, una sección de notas y la opción de firmar cada documento creado con nombre y

título del apunte. Acorde a todas estas propuestas aunado a un desarrollo rápido siempre dependiente de las opiniones y accesibilidad del usuario la metodología implementada fue metodología ágil[4], donde en todo momento se consideró:

- Las personas y las interacciones antes que los procesos y las herramientas.
- El software en funcionamiento antes que la documentación exhaustiva.
- La colaboración con el cliente antes que la negociación contractual.
- La respuesta ante el cambio antes que el apego a un plan.

En todo el desarrollo puede verse estos principios fundamentales aplicados. Al momento de ser la librería expuesta en el presente documento consumida por sus propios desarrolladores y usuarios externos no directamente relacionados con el código fuente para crear sus propios módulos inyectables, cada parte fue sometida a máximas pruebas de funcionalidad y utilidad siendo demostrable esto en cada **commit**, **merge request** o modificación de código subido y registrado en la plataforma **github**, utilizada como herramienta para control de versiones y forma segura del cumplimiento de la metodología ágil planteada.

3.1. Planeación de lenguajes, técnicas y tecnologías

El lenguaje de desarrollo fue Java por su fácil exportación multiplataforma, amplio rango de usuarios[7], estandarización de enseñanza en la academia y bibliotecas de interfaces gráficas multiplataforma 100 % orientada a objetos. Para la interfaz gráfica se usó **Swing**[6] y **AWT**[5]; bibliotecas muy flexibles, útiles y reconocidas en el lenguaje permitiendo así que cualquier usuario con intensión de modificar o personalizar los módulos se encuentre con un entorno familiar, clásico y documentado. Para lograr un proyecto fácilmente modificable e intuitivo se usaron los patrones **SOLID**(**S**RP-**O**CP-**L**SP-**I**SP-**I**DP).

SRP: Single **R**esponsability **P**rinciple por la facilidad de desarrollo del proyecto y no especialización en ningún tema externo al modulo a programar.

OCP:Open–**C**losed **P**rinciple A fin de que la lógica matemática compleja y la interfaz de interpretación de programación avanzada quedara oculto al usuario programador final.

LSP:(**L**iskov **S**ubstitution **P**rinciple) Para mantener coherente la herencia de las clases éstas son hiper especializadas siendo así más entendible al momento de heredar que atributos le proporcionaría.

ISP:(**I**nterface **S**egregation **P**rinciple) En este apartado se buscó fortalecer el mismo concepto de sub-modularizacion y super-especialización de cada objeto, evitando dejar métodos heredados sin sobrescribir o llamadas a métodos sin utilidad por herencias innecesarias.

IDP:(**D**ependency **I**nversion **P**rinciple) La principal metodología de diseño que dará todo su poder al motor matemático y su facilidad de interacción a nuevas unidades u operaciones reside en la inyección de dependencias que tiene como principio esta metodología técnica, la implementación de clases abstractas base las cuales serán las únicas variables que manejara directamente el motor creando un ambiente abstracto robusto y flexible[11].

4. Desarrollo

4.1. Base matemática del programa

El primer paso en el desarrollo del programa fue definir los objetos abstractos a usar como padres o plantillas genéricas para su posterior uso en el motor matemático. Las clases desarrolladas fueron:

- **Unidades Matemáticas:** Clase abstracta para definir un nuevo tipo de unidad en el programa, consiste en el dato que representa una forma cuantificable en las matemáticas y pueden por tanto desarrollar acciones como ser definido con sus variables específicas, ser visualizado en alguna forma gráfica, realizar operaciones matemáticas o ser resultado de alguna función matemática. Una unidad matemática está compuesta por:
  - Nombre del tipo de unidad (String del constructor).
  - Símbolo identificador (Char del constructor).
  - Nombre de la categoría que pertenece (String del constructor).
  - Un objeto que represente su valor y pueda ser retornado (Object primitivo del constructor).
  - Método que defina qué ventana gráfica necesita renderizar para ser creado y recibir los parámetros que usará. (método void abstracto heredado para crear un JFrame que represente la entrada de datos).
- **Operaciones Matemáticas:** Son el segundo tipo de dato en el programa, representan las operaciones a realizar entre dos o más unidades, son el núcleo dinámico donde al ser heredadas por el programador que deseé implementar sus propias operaciones le pedirá sobrescribir los métodos necesarios para su explicación, visualización y resolución. Una operación matemática está compuesta por:
  - Nombre del tipo de operación (String del constructor).
  - Símbolo identificador (Char del constructor).
  - Breve descripción de la operación (String del constructor).
  - Descripción detallada de la operación (String del constructor).
  - Prioridad según la jerarquía matemática (1:Sumas y Restas, 2: Multiplicaciones y Divisiones, 3:Potencias y raíces, 4:Paréntesis) (Int del constructor).
  - conLlave, booleano que representa si es una operación con paréntesis u operación-función. (Booleano del constructor) Ejemplo: ***vectorMagnitud(*****vecA+vecB+...VecN*****)**.
  - Método ***calcularOperacion()***, con N parámetros tipo UnidadMatematica de entrada y un retorno tipo **UnidadMatematica**. (Método abstracto heredado donde se implementara la lógica necesaria para resolver la operación).
  - Método definir ***TipoDeOperandoscorrectos()*** con salida tipo **String[]** que espera por retorno el "Nombre del tipo de unidad"que pueden usarse con este operando. (Método abstracto heredado donde se define si la entrada de datos pasada coincide con los datos esperados).

- **Funciones Matemáticas:** Intermedio entre la operaciones matemáticas y unidades matemáticas. Toma como entrada N parámetros especificados por el usuario desarrollador. El tipo de los N parámetros puede ser de cualquier objeto hijo de la clase unidad matemática. Su proceso de implementación consiste en sustituir métodos especificados por la plantilla padre y retornando otro objeto tipo unidad matemática el cual será en ultimo paso insertado en la operación general. Una función matemática está compuesta por:
  - Nombre del tipo de función (String del constructor).
  - Símbolo identificador (Char del constructor).
  - Descripción detallada de la función (String del constructor).
  - Método ***calcularOperacion()***, con N parámetros tipo UnidadMatematica de entrada y un retorno tipo UnidadMatematica (Método abstracto con salida tipo UnidadMatematica donde se especifica la lógica de resolución de la función).
  - Método ***llamarFuncionMatematica()*** que defina qué ventana gráfica necesita renderizar para ser creado y recibir los parámetros que usará (método void abstracto heredado para crear un JFrame que represente la entrada de datos).

Operación-función no es igual a las funciones matemáticas, una función matemática solicita N parámetros de X tipo de datos, una operación con paréntesis solicita 1 sub-operación con resultado en un único tipo de dato: (funcion 1 no es igual a la operacion 2).

$$\boldsymbol{hipotenusa(CatetoA,CatetoB),} \tag{1}$$

$$\neq$$

$$(4\vec{a}x + 6\vec{a}y + 3\vec{a}z) - (2\vec{a}x + 1\vec{a}y + 6\vec{a}z). \tag{2}$$

Existen clases desarrolladas como **EstandarNparamsJDialog.java** creadas para ser utilizadas en los métodos que piden desarrollo de interfaces gráficas si el usuario no conoce las librerías **Swing** o **AWT**.

4.2. Base/motor matemático del programa

El concepto de motor matemático hace referencia a una clase encargada únicamente en la resolución de una operación general dividida en 4 fases de resolución matemática. El motor matemático debe contar con la capacidad de poder resolver operaciones según el orden de jerarquía, sin importar la longitud de la operación, sin importar los operandos ni los operadores y ser robusto al retornar errores y el causante de estos. Las técnicas de programación implementadas en este proceso principal del proyecto son algoritmos de recursividad al estilo **Divide and Conquer**, objetos tipo **ArrayList** para

aplicar los conceptos de **Stack** y la implementación de un flujo de resolución el cual consiste en 4 pasos de prioridad cada uno dependiente del anterior con cinco banderas de error las cuales representan un tipo de error diferente dependiendo el proceso que fue insatisfactorio.

Pasos del motor: Al instanciar la clase **OperacionGeneral** se solicita el valor **List<ObjetoMatematico>** el cual consiste en una lista de entrada que contendrá todos los objetos tipo **OperacionMatematica** y **UnidadMatematica** que conforman nuestra operación a resolver. Al crear un nuevo objeto tipo **OperacionMatematica** solicita por el constructor padre el tipo de orden jerárquico al que pertenece el cual será definitorio para el siguiente proceso.

- Orden de operación 4: En este primer orden se encontraran todos los objetos los cuales son operaciones con paréntesis, en esta categoría entran las operación-función encapsuladas en paréntesis (3+2) u operaciones que solicitan un parámetro de entrada y retornan una unidad matemática como el ejemplo 3:

$$\textbf{\textit{vecMagnitud(}}3\vec{a}x + 2\vec{a}y + 5\vec{a}z). \tag{3}$$

  Esto se guarda en el ArrayList **ObjetosMatematicosPrimerOrden** los resultados de las operaciones y pasando sin modificación cualquier unidad matemática u operación de orden jerárquico menor a 4. Una vez resuelto todo este primer proceso el resultado almacenado de esta nueva operación sin jerarquía 4 será pasado como entrada a la siguiente parte del flujo.
- Orden de operación menor a 4: En esta parte del proceso se hace uso del método ***resolverOrdenN()*** el cual es un método genérico para resolver todo tipo de operación menor a 4 o en otras palabras toda operación que no haga uso de paréntesis sino operandos de lado izquierdo y operandos de lado derecho. En esta parte del flujo la función es llamada 3 veces para la resolución de los ordenes de tipo 3 (potencias,raíces...), 2 (multiplicaciones,divisiones), 1 (sumas, restas, unidades sueltas finales).
  operación tipo 3:

$$5^2,$$

  operación tipo 2:

$$(4\vec{a}x + 6\vec{a}y + 3\vec{a}z) * 25,$$

  operación tipo 1:

$$(4\vec{a}x + 6\vec{a}y + 3\vec{a}z) - (2\vec{a}x + 1\vec{a}y + 6\vec{a}z),$$

  operación tipo Resultado:

$$(2\vec{a}x + 64\vec{a}y + 2\vec{a}z).$$

5. Interfaz gráfica

Una vez implementada la base lógica del programa fue necesario crear la interfaz gráfica. Ésta debía ser modular, personalizable y digerida para que

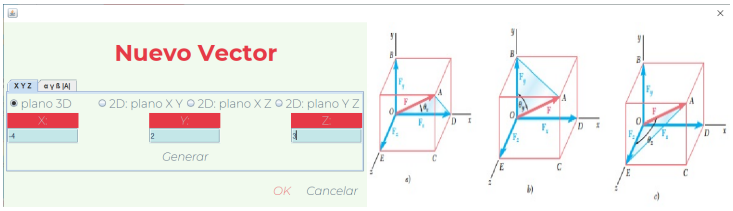


Fig. 1. Ejemplo de un panel de creación personalizado para vectores.

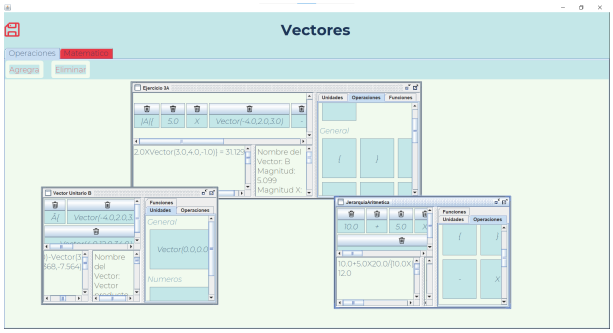


Fig. 2. Panel de operaciones con 3 operaciones generales resueltas.

cualquier usuario programador sin mucha experiencia en el lenguaje pueda modificarla sin necesidad de un profundo conocimiento de las bibliotecas **Swing** o **AWS**. Por ello, los procesos de renderizado son privados. Ejemplos de estos son: **OperacionesGenerales** y **ProcedimientosGenerales**. Sin embargo, las secciones relacionadas con la creación de nuevas **UnidadesMatematicas**, **OperacionesMatematicas** o **FuncionesMatematicas** son públicas y abstractas, lo que permite su posterior sobre-escritura a fin de personalizarlas. Esto último, si se tienen los conocimientos en el lenguaje (Fig.1), y si no es el caso, se pueden importar las clases base de acceso público, completamente compatibles con la interfaz y cumplen las necesidades genéricas de entrada y salida de datos de forma gráfica.

5.1. Panel de operaciones

Representa la principal parte gráfica del programa (Fig.2). Aquí se lleva el proceso de creación de nuevas y múltiples operaciones generales, selección de la operación, creación de la ecuación a resolver, resultados de la operación, entre otras funciones. Este panel de operaciones es un espacio de creación de sub-ventanas dentro del mismo programa las cuales se segmentan en las siguientes funcionalidades:

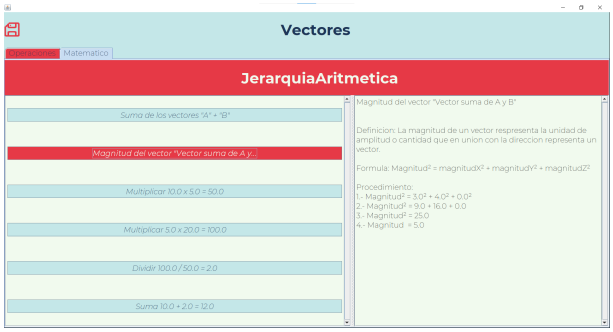


Fig. 3. Panel de procedimiento de una operación general llamada **"JerarquiaAritmetica"**.

Panel Inspector	En él se encuentran las unidades, operaciones y funciones creadas en el programa divididas dependiendo su categoría matemática y subcategoria de tema especificado en el momento de creación.
Panel Operaciones	Panel con funcionalidad *Drag and Drop* el cual establece la operación matemática a resolver. Permite hacer modificaciones como el orden de la operación, remover objetos o seleccionar el objeto enfocado.
Panel Detalles	Representa el objeto matemático seleccionado desde el panel operaciones, aquí se da una breve descripción del objeto. En caso de no haber sido seleccionado ningún objeto será mostrado por defecto los detalles del resultado de la operación.
Panel Resultado	Muestra en una cadena (*string*) auto-generada el proceso matemático a realizar y su respectiva solución.

5.2. Panel de procedimiento

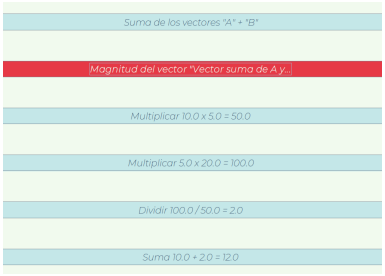
En él, se encuentra todo el procedimiento del motor matemático para llegar al resultado final de la operación general seleccionada en el panel de operaciones (Fig.3).

5.3. Paneles modulares genéricos

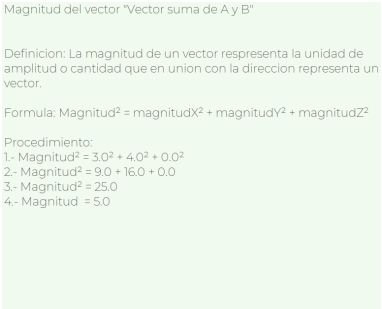
En esta categoría de paneles entran los extra de personalización como los paneles de Pizarra (Fig. 5a), notas (Fig. 5b) y graficación (Fig. 5c).

6. Sistema de apuntes

El sistema de apuntes consiste en la forma que el programa tiene para guardar los archivos generados y poderlos cargar desde su interfaz de entrada



(a) Índice de procedimientos: Permite seleccionar la operación que se visualizará en el panel derecho



(b) Descripción: Panel de descripción detallado con el texto auto-generado con base en la plantilla proporcionada por el programador.

Fig. 4. Paneles izquierdo y derecho de figura 3.

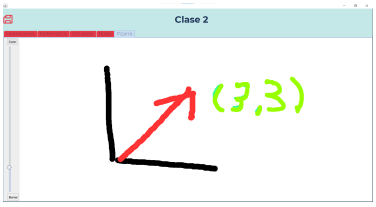
(Fig. 6). Esto tiene como objetivo facilitar la creación de material didáctico tanto para maestros o alumnos que quieren compartir la lección del día en un documento estandarizado con las operaciones realizadas, la calculadora que la hizo, explicación de cada procedimiento hecho, notas, algún dibujo y hasta una representación gráfica todo en un solo archivo.

7. Casos de prueba

Para el plan de pruebas se desarrollaron un total de 30 pruebas divididas entre casos de uso con vectores 3D, números reales, operaciones de diferentes tipos de unidad matemática en conjunto y operaciones con jerarquía de resolución. La metodología para la realización de pruebas se apegó a los principios **Big Bang integration testing**[9] siendo evaluado no únicamente la parte matemática sino la funcionalidad, uso del programa y el conjunto entero terminado de un ejecutable listo para la resolución y explicación de problemas matemáticos. Una vez que el motor logró resolver correctamente todas las operaciones, comprobar la funcionalidad de interfaz y la posterior explicación del procedimiento, fueron seleccionadas las más importantes con base en criterios de longitud, recursividad de operación (cantidad de sub-operaciones) y uso de diferentes unidades matemáticas inyectadas entre ellas, las principales fueron las siguientes:

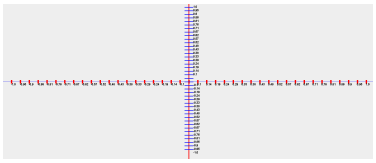
Operación con multiplicación de vectores por algún escalar para posteriormente poder realizar su resta (Op. 4), operación importante que implica el uso dinámico de 2 unidades inyectadas de forma independiente y la resolución por un orden de jerarquía. Fuente: Problema 13 primera parte inciso A libro "Métodos matemáticos para físicos"[8].

$$5{,}0 * (-4a\vec{x} + 2a\vec{y} + 3a\vec{z}) - 2{,}0 * (3a\vec{x} + 4a\vec{y} - 1a\vec{z}). \tag{4}$$



(a) Panel Pizarra.

(b) Panel Notas.



(c) Panel Gráficos

Fig. 5. Paneles modulares.

$$(-20\vec{ax} + 10\vec{ay} + 15\vec{az}) - (6\vec{ax} + 8\vec{ay} - 2\vec{az}),$$
$$(-26\vec{ax} + 2\vec{ay} + 17\vec{az}).$$

Este resultado es verificado en la Fig. 7.

Caso de pruebas con una operación tipo operación-función, la resolución de este problema implica el correcto funcionamiento de la jerarquía 4 y las jerarquías $N < 4$ en conjunto (Op.5). Fuente: Problema 13 primera parte inciso B libro "Métodos matemáticos para físicos"[8].

$$|5{,}0 * (-4\vec{ax} + 2\vec{ay} + 3\vec{az}) - 2{,}0 * (3\vec{ax} + 4\vec{ay} - 1\vec{az})|\,, \tag{5}$$
$$|(-20\vec{ax} + 10\vec{ay} + 15\vec{az}) - (6\vec{ax} + 8\vec{ay} - 2\vec{az})|\,,$$
$$|(-26\vec{ax} + 2\vec{ay} + 17\vec{az})|\,\sqrt{26^2 + 2^2 + 17^2} = 31{,}129.$$

La Fig. 8 muestra la verificación de este resultado.

Operación con uso de una alta capacidad de jerarquización y resolución de múltiples sub-operaciones en el orden correcto Op.6. Problema de caso útil para comprobar la salida del proceso matemático como fuente de estudio para el usuario no programador. Fuente: problema aleatorio generado específicamente para el motor.

$$10 + 5 * 20/10\,|(2\vec{ax} + 3\vec{ay}) + (\vec{ax} + \vec{ay})|\,, \tag{6}$$
$$10 + 100/10\,|(3\vec{ax} + 4\vec{ay}))|\,,$$
$$10 + 100/10 * (5),$$
$$10 + 100/50,$$
$$10 + 2 = 12.$$

Este resultado es comprobado con la Fig.10.

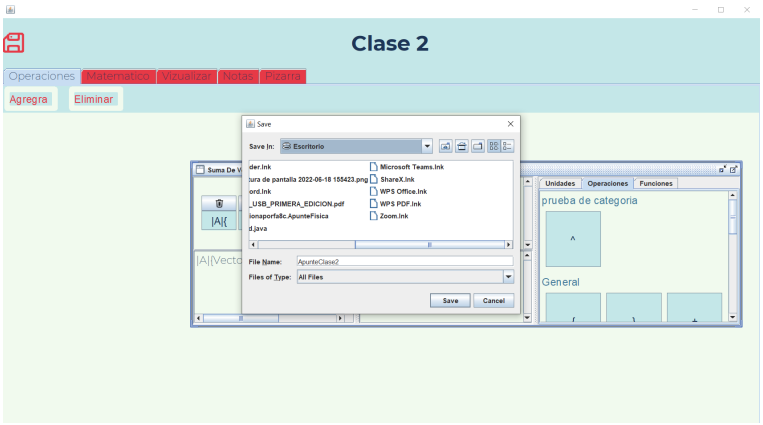


Fig. 6. Guardado de un nuevo apunte con todos los datos creados.

Fig. 7. Panel de operaciones de la ecuación 4.

8. Conclusiones

En este artículo se presentó el desarrollo de una biblioteca matemática para su fácil implementación o adaptación a cualquier usuario en la materia de programación o matemáticas, cumpliendo con las características solicitadas en las metas a cumplir, compatible con cualquier unidad desarrollada por usuarios programadores, reutilizable con módulos externos de otros desarrolladores, además modificable, e implementable para cualquier persona sin experiencia en el lenguaje. Posee una interfaz gráfica amigable a los usuarios finales. En la evaluación de la aplicación, se obtuvo un buen rendimiento en todos los problemas resueltos. El motor matemático con inyección de módulos externos e interfaz gráfica modular se encuentra lista en su primera versión para poder ser usada y distribuida en un ambiente de producción muy útil para las aulas de clase, alumnos autodidactas y grupos de estudio que buscan una

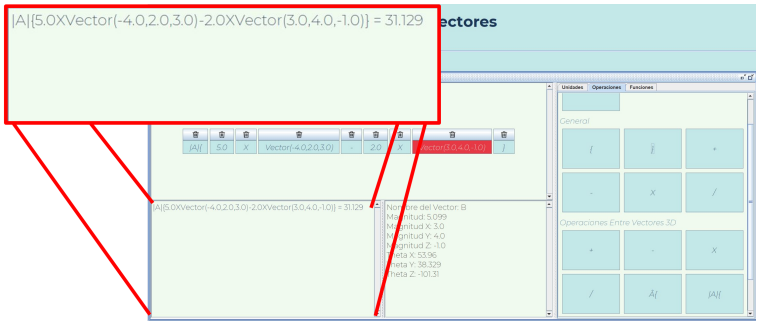


Fig. 8. Panel de operaciones de la ecuación 5.

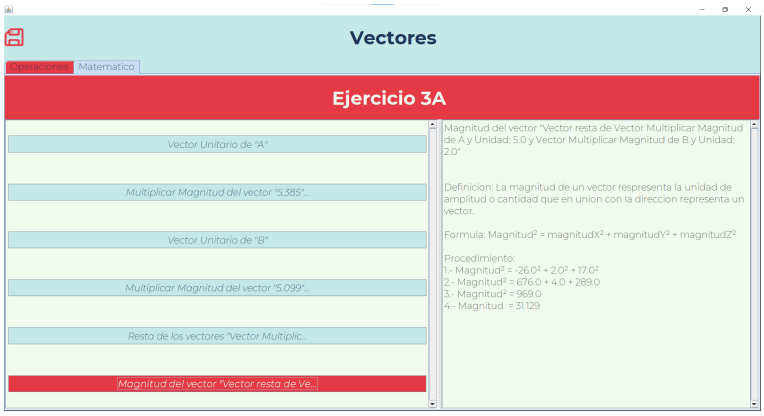


Fig. 9. Panel de procedimiento de la ecuación 5.

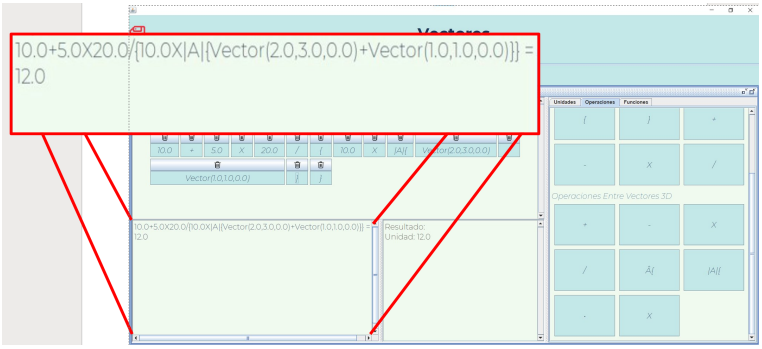


Fig. 10. Panel de operaciones de la ecuación 6.

rápida automatización de procesos manuales. El proyecto mostrado aquí aun se encuentra en su primera versión, pero se constata su gran potencial para popularizarse y albergar un diversidad de módulos que pueden ser creados por

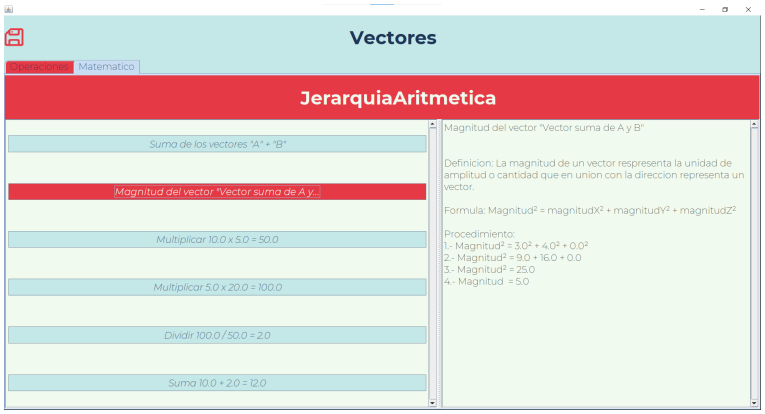


Fig. 11. Panel de procedimiento de la ecuación 6.

la comunidad de usuarios programadores. Esto permitiría su crecimiento y ser ofrecida como un sistema *Open Source*. Finalmente otra área de oportunidad que puede resultar atractivo entre desarrolladores interesados en implementar sus propios módulos y compartirlos a la comunidad seria migrar el proyecto de un ambiente de escritorio a una pagina web, adaptar las vistas mostradas en el presente documento a una forma mas acorde a los estándares de diseño de hoy en día y crear un sistema de desarrollo y red de difusión de módulos entre autores y usuarios no programadores. La lógica del proyecto se encuentra completamente preparada para crecer y su mayor potencial se verá alcanzado una vez que la cantidad de módulos creados pueda equiparar a los grandes programas de pago o lenguajes de programación privados explicados anteriormente en la sección 2. En términos de características, utilidades y funciones, el motor cumple con ser libre para su uso ilimitado, explicaciones paso a paso personalizadas por el usuario, entrada, salida o representación de datos según las necesidades y finalmente implementado en un lenguaje conocido y fácil de aprender como Java, representando así una gran ventaja en conjunto ante los trabajos citados en la sección 2.1, 2.2 y 2.3.

Repositorio con los códigos fuente, implementaciones y recursos gráficos listos para ser compilados, modificados, importados o ejecutados se encuentra en Github[1], igual que el motor matemático con inyección de dependencias dinámicas (Motomaticas).

Referencias

1. Matlab, `https://www.mathworks.com/products/matlab.html`
2. Symbolab math solver - step by step calculator, `https://www.symbolab.com/`
3. The world's favorite, free math tools used by over 100 million students and teachers, `https://www.geogebra.org/`

[1] `https://github.com/hamletSolanoD/Motomaticas`

4. ¿qué es la metodología ágil?, https://www.redhat.com/es/devops/what-is-agile-methodology
5. Package nbsp;java.awt (Jun 2020), https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html
6. Package nbsp;javax.swing (Jun 2020), https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html
7. Tiobe index (Jun 2022), https://www.tiobe.com/tiobe-index/
8. Arfken, G., Weber, H. J.: Mathematical methods for physicists. Harcourt Academic Press (2003)
9. Hanh, V. L., Akif, K., Traon, Y. L., Jézéque, J.-M.: Selecting an efficient oo integration testing strategy: an experimental comparison of actual strategies. In: European Conference on Object-Oriented Programming. pp. 381–401. Springer (2001)
10. Reingold, E. M.: A comment on the evaluation of Polish postfix expressions. The Computer Journal, vol. 24, no. 3, pp. 288–288 (01 1981) doi: 10.1093/comjnl/24.3.288
11. Yang, H. Y., Tempero, E., Melton, H.: An empirical study into use of dependency injection in java. In: 19th Australian Conference on Software Engineering (aswec 2008). pp. 239–247 (2008) doi: 10.1109/ASWEC.2008.4483212

Study of Decentralized RF and LiFi Networks as a Complement to Congestion in Centralized Networks

ISSN 1870-4069

Gerardo Hernández Oregón, Jorge Enrique Coyac-Torres, Mario Eduardo Rivero

Instituto Politécnico Nacional,
Centro de Investigación en Computación, Mexico City,
Mexico

Abstract. Congestion in current centralized networks is an increasing problem with the number of devices worldwide. Because of the above, enormous efforts have been made to provide new alternative solutions to this problem, such as decentralized networks. This work proposes a novel Peer-to-Peer (P2P) structure through Radio Frequency (RF) and Light-Fidelity (LiFi) technologies, performing a Discrete Event Simulation (DES) of the transmission times concerning the parameters and operation of each of these technologies. Besides, this research presents the impact of nodes and bandwidths variations for *Upload* and *Download* for this type of P2P network and their possible applications. Finally, the comparison between centralized networks and decentralized networks is analyzed.

Keywords: Peer-to-peer, Ligth-Fidelity, Radio-Frequency, descentralized, networking.

1 Introduction

According to Cisco, it is predicted that more than 500 billion devices will be connected to the Internet by the year 2030 [10]. All this is due to the rise of new technologies such as Internet of Things (IoT) and its increasingly daily use, which can range from process automation to the use of wearables that allow the monitoring of biomedical signals [9]; not to mention the multiple guides and technical manuals that exist for the implementation of this technology in almost any field [8].

Due to the reasons mentioned earlier, the communications between all these devices and sensors will have to be carried out mostly without human intervention, this may cause the rise of P2P active node structures and their variants (Machine-to-Machine (M2M), Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and others), which can better deal with the exorbitant amounts of data created and seem to complement centralized networks nicely.

For a long time, Radiofrequency networks have been studied by many researchers in the communications area; therefore, their areas of use and parameters are largely known. However, LiFi is a technology in the development process and constantly exploring its main niches, obtaining its areas of opportunity in hostile networks to radio frequency. Networks in which the nodes are static and those that require dedicated links without wiring. Thanks to its characteristics of high security in the physical and data link layer (OSI model), high bandwidth (unlicensed), high data density, and adaptability [6], LiFi is positioned as a technology that is perfectly compatible with P2P environments in the propagation of information of interest through all the nodes of the network.

2 Network Architecture

The architecture in data networks is related to structure, both, logical and physical organization of the components of a telecommunications network. Generally, architectures can be measured by standard parameters such as bandwidth, transmission speed, storage or processing capacity, the technologies used for their operation, the network topologies used, and how nodes interact on the net. Due to this last parameter, two architectures with a significant impact on current networks have traditionally been used: centralized architectures (Client-Server) and decentralized architectures (Peer-to-Peer) [7].

In recent years, the benefits of both technologies have been used to create hybrid architecture networks for applications such as adaptive video streaming [2], cloud applications, or multi-agent optimizations [3]. Figure 1 shows an example of the use of this environment.

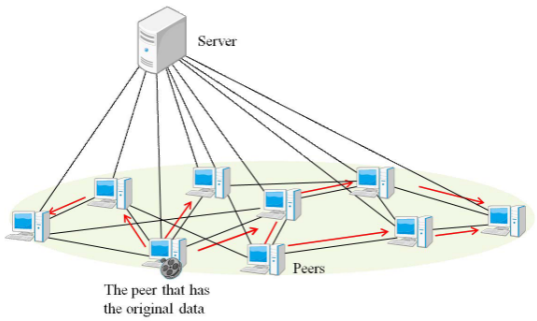


Fig. 1. P2P streaming environment.

2.1 Centralized Networks

Centralized architectures are characterized by having a main server from which other nodes obtain information. For these reasons, the main server often must

present robustness of processing, storage, and bandwidth to supply the entire network. Otherwise, the phenomenon of congestion is shown, which in extreme cases can damage the entire network; thus, the server is vital for the centralized model.

On the other hand, centralized entities allow information control to be concentrated in a specific group of servers that can respond to transactions and are ideal for environments such as banking, government, universities, and others.

2.2 Descentralized (P2P) Network

Another computer architecture contains the Peer-to-Peer (P2P) model, which has become relevant in recent years. This architecture promotes the activity of all the nodes in the network since it segments the packets and allows each node to behave as a client and server, obtaining the segments (chunks) that it needs and sharing those it has.

Such as centralized networks, there is a client and a server. P2P networks have two types of nodes: Leechers and Seeds. Leechers are those nodes that do not have all the chunks, while Seeds are those nodes that have all the information available in the network (Leechers and Seeds [5] can share their information).

2.3 P2P Networks for RF and LiFi Technologies

In this work, the simulations of the technologies described in the preceding paragraphs are shown, proposing a centralized network that will have the same parameters as the decentralized networks and whose simulation will differ concerning the latter in obtaining information by the present nodes on the network by getting a server all over the network.

For the P2P-RF network, a maximum range network will be considered (all nodes in the network could potentially connect to each other). While for the P2P-LiFi network, its functionality is described in the simulation in section *3.1*. It is important to highlight that one of the essential points of this research is present in the visualization, approach, and simulation of P2P-LiFi networks, the implementation proposal for places with static or semi-static nodes, and a comparison with P2P-RF networks and centralized.

3 Results

This section details the results obtained in the simulations by discrete events of the system (DES) carried out for the technologies specified in subsection *2.3*. Also, the general considerations of the proposed scenarios are pointed out. Finally, the corresponding results to the dispersion of files across the network and the impact on the variation of the number of nodes in these environments were analyzed, and they are presented in the following sections of this work.

3.1 Node Description for Static and Semistatic Nodes

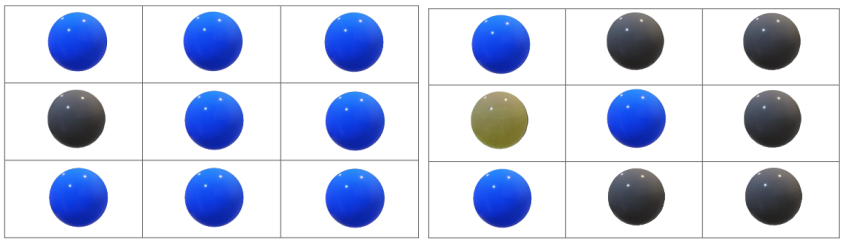
Since LiFi technology depends on unique characteristics for its communication, such as its field of vision and its line of sight, the ideal environments involve static or semi-static nodes, which is why these types of applications are excellent for places such as hospitals, museums, offices, industry, and classrooms.

For the proposed experiments, a centralized scenario is proposed through an RF network with a maximum coverage radius and two decentralized scenarios using the P2P architecture with LiFi and RF technologies, whose considerations are specified in Table 1.

Table 1. General considerations for P2P static and semistatic scenarios.

Parameters	**RF**	**LiFi**
Max. Upload connections	1	1
Max. Download connections	4	4
Coverage radius	Max.	LOS
Noise and intereferences	not considered	not considered
Peer connection improvement (DES)	Yes	Yes

The maximum data upload and download connections are used according to the classic model of some P2P networks.



(a). Maximum range P2P-RF network **(b).** LOS range P2P-LiFi network

Fig. 2. Description of a connection through proposed P2P-RF and P2P-LiFi networks.

Figure 2 describes that any node has a sufficient coverage radius to connect to another within the RF network. Meanwhile, in the case of the LiFi network, the nodes can only connect with those available in their Line Of Sight (LOS) and located at the four cardinal points.

Furthermore, Figure 2-a shows the maximum range P2P-RF network in which a node (black node) can connect to any other node within the network (blue nodes); then, maximum node reach allows all nodes within the network to be

reached by the node that needs to connect. By another hand, Figure 2-b shows the LOS range P2P-LiFi network in which a node (red node) can connect to any other node within its LOS (blue nodes). It should be noted that some nodes are not reachable (black nodes) by the node that tries to obtain the information.

It is true that there are well-identified characteristics in this technology, such as high data density, bandwidths of up to 10Gbps (unlicensed), and transmission only under its Field Of View (FOV) aligned to the LOS of the same device, for which there is already a large number of works and prototypes that take into account all these characteristics in P2P environments [1,4]. However, LiFi devices are yet in the standardization process, study use cases, and constant development. Figure 3 is a clear example of the ways in which LiFi devices can be presented in different areas. This LiFi device could be a prototype for the physical realization of the P2P-LiFi network proposed in this work.

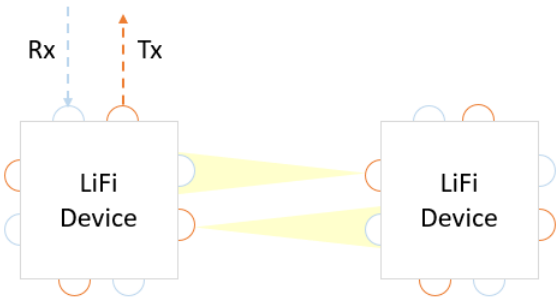


Fig. 3. DES implemented for the case of 8 nodes in the centralized, P2P-RF, and P2P-LiFi technologies.

3.2 DES for Centralized and Descentralized Environments

DES processes allow systems to be recreated through their characteristics and behaviors over time. At a first glance, the simulation for this work recreated the behavior of P2P networks regarding connection, disconnection, election, births and deaths of nodes and adapted these aspects to the proposed scenarios.

P2P systems have been studied daily from Markovian models using queuing systems to describe the number of average Seeds and Leechers and the service times in the system. The previous is directly linked to the dynamics of classic P2P systems according to the constant movements, connections, and disconnections through the nodes present in the network. However, the main objective of the case study of this article is not the dynamics of the movement of the nodes over

time; since it is intended for communication networks in which the nodes are static or semi-static.

Therefore, the number of Leechers and Seeds through time remains constant due to the nature of the proposed technologies. Consequently, the object of study in this type of system is the file's download time through all the nodes present in the network, which is calculated through Discrete Events Simulation (DES) as described in Algorithm 1.

Data: $nodes, \mu, radius$
Result: $timesimulation$
$timesimulation \leftarrow 0$;
$seeds \leftarrow 1$;
$leechers \leftarrow 1$;
$Add(seed)$;
$seeds \leftarrow seeds + 1$;
$i \leftarrow 0$;
while $i < nodes - 1$ **do**
  $Add(leecher)$;
end
while $seeds < nodes$ **do**
  $conversions \leftarrow leechers2seeds(timesimulation)$;
  $seeds \leftarrow seeds + conversions$;
  $event \leftarrow event.getfront()$;
  $timesimulation \leftarrow event.time()$;
end

Algorithm 1: Discrete Event Simulation (DES) algorithm performed for P2P-RF and P2P-LiFi networks.

The performed DESs obtain the total time in which a file is shared through the network of nodes, taking into account the nature and characteristics of each technology to connect and transfer information. The previous algorithm also shows that each time an event is fulfilled, Leechers that for that time t could be converted into Seeds are updated. Finalizing the simulation when all the nodes have been converted into Seeds.

3.3 Information Sharing over Time and the Impact of Network Node Variation

This part of the work shows the relevant results of this research. These results are shown through surfaces that reflect the behavior of the information transmission times along the different bandwidths shown in Table 2. Following what was stated in the previous lines, *Figure 4* shows in the first instance that although the variation of the data download bandwidth (c) also impacts the total transmission time.

The upload bandwidth of the data (μ) represents a bottleneck for the system, which also causes an impact on the sharing times of the system for the given criteria. One thing to confirm in the graphs is that the higher the bandwidth, the shorter the time it takes for the information to propagate through the nodes.

Table 2. Number of nodes and bandwidths utilized in DES.

Parameter	**Values**
μ	[0.001, 0.002,0.003,0.004,0.005]
c	[0.01, 0.02,0.03,0.04,0.05]
Nodes	[4,8,12,16,20,24,28,32,36]

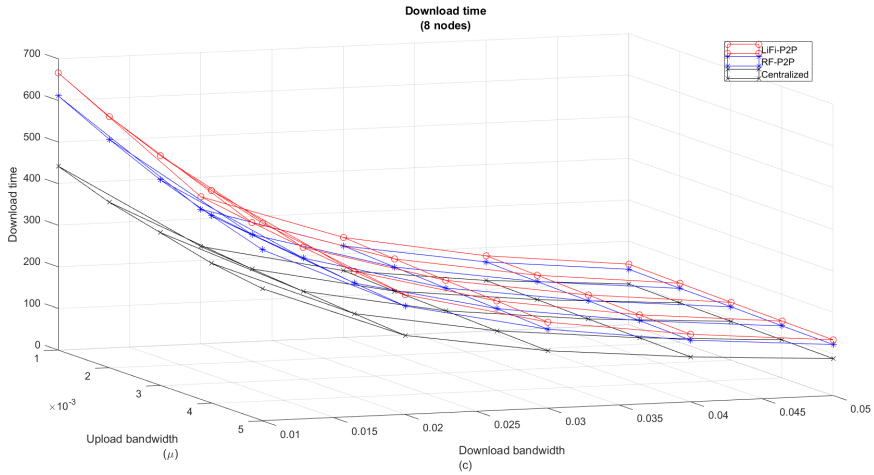
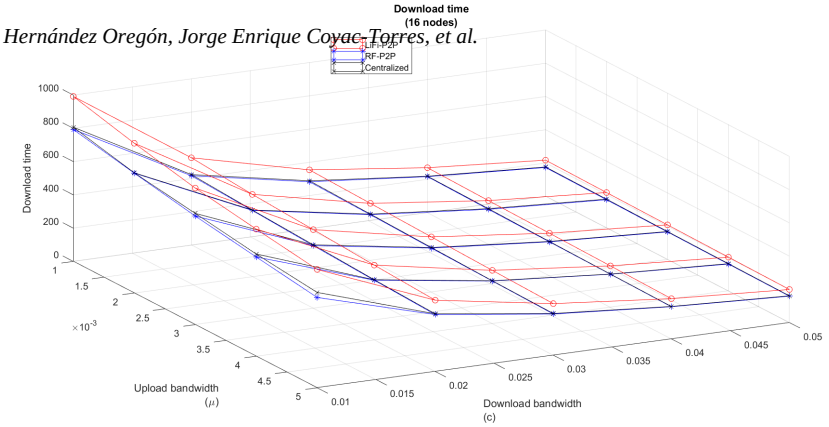


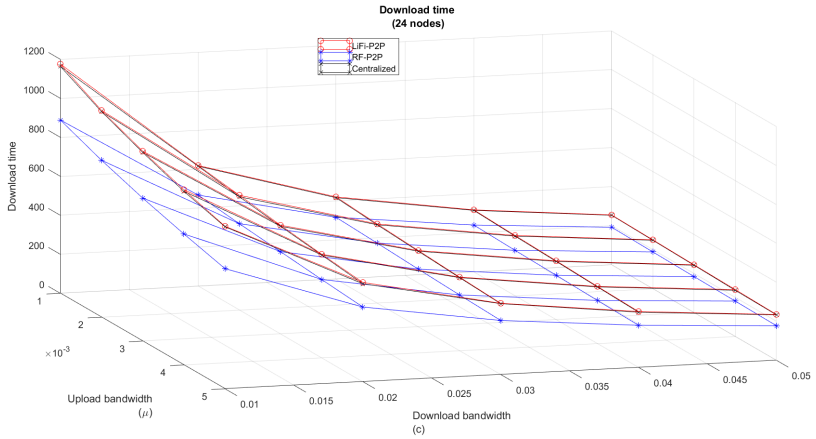
Fig. 4. Performed DES for the case of 8 nodes in centralized, P2P-RF, and P2P-LiFi technologies.

In addition to the variation of the bandwidths μ and c, in *Figure 5* we can look at the behavior of these networks from another perspective due to the number of nodes that the network takes into account. Simulation of the proposed system. The centralized technology obtains better performance for the case of few nodes, while, as the number of nodes increases, the P2P networks begin to work better; this is because the centralized network becomes congested, and the decentralized networks begin to spread the information through Leechers constantly turning into Seeds in the system.

For the case of the P2P-RF network, it can be seen that a connection with the centralized network is achieved at 16 nodes (*Figure 5-a*). Meanwhile, the breaking point for the P2P-LiFi network is located at 24 nodes (*Figure 5-b*).



(a). Performed DES for the case of 16 nodes in centralized, P2P-RF, and P2P-LiFi technologies.



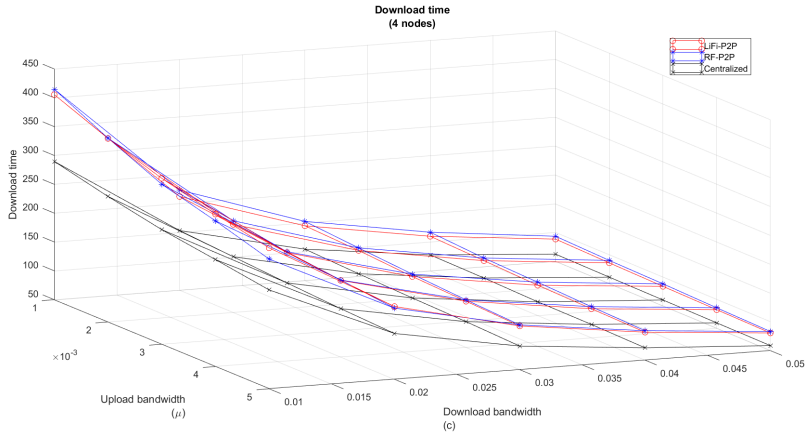
(b). Performed DES for the case of 24 nodes in centralized, P2P-RF, and P2P-LiFi technologies.

Fig. 5. Breakpoints of P2P networks concerning centralized network.

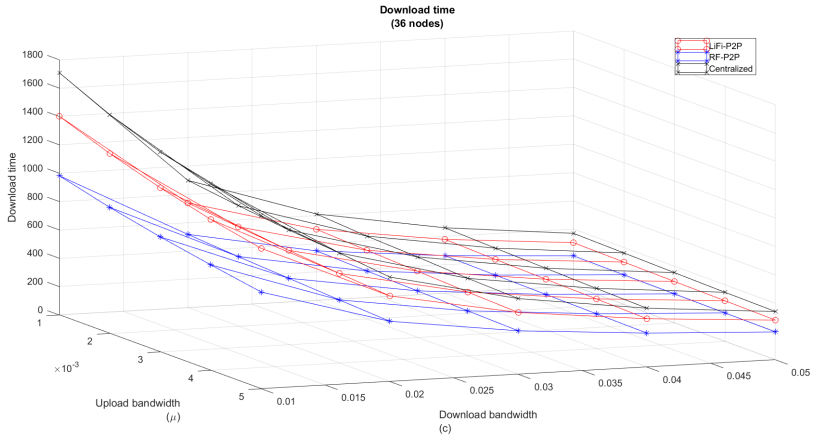
As discussed in previous paragraphs, P2P networks are rarely affected by a plethora of nodes. On the contrary, they generally show better performance in environments with a large number of nodes. The *Figure 6* confirms the previous assertion, observing that for 36 nodes (*Figure 6-b*), the behavior of the P2P-RF and P2P-LiFi networks show shorter information sharing times than in the centralized network.

It should be noted that the P2P-RF network always presents lower transmission times than the P2P-LiFi network due to the maximum range coverage

it presents and the intrinsic nature of LiFi technology to transmit in its line of sight.



(a). Performed DES for the case of 4 nodes in centralized, P2P-RF and P2P-LiFi technologies.



(b). Performed DES for the case of 36 nodes in centralized, P2P-RF and P2P-LiFi technologies.

Fig. 6. Impact of node variation in P2P and centralized networks.

4 Conclusions

This work proposes a new approach to decentralized networks through the concept of P2P networks applied to static and semi-static networks. It also exposes the study of the impact of parameters such as number of nodes and bandwidths in the proposed environments and compares their performance with that of a

centralized network. The article describes the discrete event simulation of the proposed systems and the proposed comparisons, and reports the results obtained.

P2P-RF and P2P-LiFi networks presented a better performance with a more significant number of nodes than the centralized network, showing that this kind of network can be an excellent complement to congestion as we could see into performance comparation between P2P and centralized networks(essential part from this paper). On the other hand, it may be affirmed that due to the different operation of each technology there was a variation in the times for data sharing, in which, LiFi always was maintained with greater times than RF, for above, future work will address standardized bandwidths for each technology in order to have a new differentiator between these proposed P2P networks and thus be able to see the opportunity areas for LiFi over RF.

References

1. Aleksieva, V., Valchanov, H., Dinev, D.: Comparison study of prototypes based on lifi technology. In: 2019 International Conference on Biomedical Innovations and Applications (BIA). pp. 1–4. IEEE (2019)
2. Ghareeb, M., El-Rody, R., Cheaib, A., Raad, M.: Client/server and peer-to-peer hybrid architecture for adaptive video streaming. In: 2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15). pp. 1–6. IEEE (2015)
3. Hale, M.T., Nedić, A., Egerstedt, M.: Cloud-based centralized/decentralized multi-agent optimization with communication delays. In: 2015 54th IEEE Conference on Decision and Control (CDC). pp. 700–705. IEEE (2015)
4. Mat, N., Rashidi, C., Aljunid, S., Endut, R., Ali, N.: Enrichment of wireless data transmission based on visible light communication for triple play service application. In: AIP Conference Proceedings. vol. 2203, p. 020066. AIP Publishing LLC (2020)
5. Qiu, D., Srikant, R.: Modeling and performance analysis of bittorrent-like peer-to-peer networks. ACM SIGCOMM computer communication review 34(4), 367–378 (2004)
6. Sahrawat, P.K.: Li-fi: future of wireless comunication. In: Proceedings of national conference on innovative trends in computer science engineering (2015)
7. Sakashita, S., Yoshihisa, T., Hara, T., Nishio, S.: A data reception method to reduce interruption time in p2p streaming environments. In: 2010 13th International Conference on Network-Based Information Systems. pp. 166–172. IEEE (2010)
8. Torres-Restrepo, L., Martínez-Rebollar, A., González-Mendoza, M., Estrada-Esquivel, H., Vargas-Agudelo, F.: Method for introducing iot project development using free software tools. Research in Computing Science (2020)
9. Zacatelco Barrios, L.B., Tovar Corona, B., Pindter Medina, J.: Wearable para monitoreo de ritmo cardíaco y actividad electrodérmica. Research in Computing Science (2020)
10. Zikria, Y.B., Ali, R., Afzal, M.K., Kim, S.W.: Next-generation internet of things (iot): Opportunities, challenges, and solutions. Sensors 21(4), 1174 (2021)

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación