Revista de la Escuela Colombiana de Ingeniería, N.° 119
ISSN 0121-5132 Julio-septiembre de 2020, pp. 65-75

Arquitecturas para el análisis de grandes cantidades de datos en tiempo real, aplicado a criptomonedas

Architectures for analyzing large amounts of data in real time, applied to cryptocurrencies

JUAN PABLO ARÉVALO MERCHÁN
Estudiante de la Maestría en Informática de la Escuela Colombiana de Ingeniería Julio Garavito.

juan.arevalo-m@mail.escuelaing.edu.co

Recibido: 25/04/2020 Aceptado: 15/05/2020
Disponible en http://www.escuelaing.edu.co/es/publicaciones_revista
http://revistas.escuelaing.edu.co/index.php/reci

Resumen

Diariamente, se generan grandes cantidades de datos en internet. El crecimiento exponencial de los dispositivos conectados y las nuevas formas de interacción humana con la tecnología proporcionan una gran fuente de información.

En este contexto, el análisis de datos en tiempo real es fundamental para tomar decisiones basadas en información actualizada. Varios factores, como la latencia, la escalabilidad, el almacenamiento, el procesamiento, la visualización y la predicción, afectan la forma en que se reciben y procesan estos datos. Diseñar una arquitectura adecuada para estos sistemas es un elemento crítico de cualquier negocio o empresa respaldada por la tecnología.

En este artículo se estudia el estado del arte de las arquitecturas de sistemas para el análisis en tiempo real de *big data* en el contexto de las criptomonedas. Se presentan primero lambda y kappa, dos arquitecturas de referencia para el análisis en tiempo real en sistemas *big data*, y luego se exploran varios trabajos en los que se describen las arquitecturas de los sistemas de análisis en tiempo real para criptomonedas.

Palabras claves: *big data*, arquitectura kappa, arquitectura lambda, *streaming*, criptomonedas, arquitecturas *big data* en tiempo real.

Abstract

Large amounts of data are being generated daily on the internet. The exponential growth of connected devices and the new ways of human interactions with technology provide a prolific information source.

In this context, real-time data analysis is essential for making decisions based on updated information. Several factors, such as latency, scalability, storage, processing, visualization, and prediction, affect how this data is received and processed. Designing a suitable architecture for these systems is a critical element of any business or social endeavor supported by technology.

This article studies the state of the art of the architectures of systems for real-time analysis of big data in the context of cryptocurrency. The paper first presents lambda and kappa, two reference architectures for real-time analysis in big data systems. It then explores several works describing the architectures of real-time analysis systems for cryptocurrency.

Keywords: big data, kappa architecture, lambda architecture, streaming, cryptocurrency, real-time big data architecture.

INTRODUCCIÓN

En el mundo actual, la generación de datos crece en una cantidad exponencial, debido a los numerosos dispositivos capaces de generarlos, desde los sensores IoT más básicos, pasando por transacciones financieras y de criptomonedas, y llegando incluso a información reservada para temas de ciberseguridad.

Durante los últimos años, el análisis de datos en tiempo real ha cobrado mucha fuerza, puesto que cada vez es más importante contar con la información más reciente para la toma de decisiones. En un mundo tan competitivo, esto se tiene que sustentar efectivamente en la información generada por las compañías, la cual tiene que estar lo más actualizada posible; si no es así, las decisiones podrían basarse en datos errados o desactualizados.

Un ejemplo típico sobre el análisis de datos en tiempo real es el aplicado en el campo de las criptomonedas. En la actualidad, estas monedas tienen un valor monetario muy grande; tanto así que plataformas creadas para hacer un seguimiento de la capitalización de diferentes criptomonedas [25] informan que a la fecha existen más de 7.800 clases de estas monedas.

Según [25, 26], las principales criptomonedas son bitcóin, ethereum, ripple, litecoin y tether. Todas presentan una condición común: la volatilidad en sus precios. Por ejemplo, el bitcóin ha pasado de un precio de US$1.000 en 2017 a US$19.000 en 2020 [25]; esto hace que exista un alto riesgo en tales inversiones, pero a su vez dicha condición vuelve muy atractivo el mercado, pues se puede llegar a ganar mucho dinero.

Existen varias teorías en cuanto a factores incidentes en los precios de estas monedas. Algunos –como [5]– dicen que este factor es determinado por la oferta y la demanda, por los sentimientos de pánico, pesimismo, escepticismo, optimismo y euforia, y por la opinión de las personas; esto hace que sea muy importante tener en conjunto toda esta información en tiempo real, procesada y lista para su utilización.

Criptomonedas como ethereum manejan cerca de 1.200.000 transacciones por día [27], lo cual demuestra la cantidad de información que se debe evaluar.

Aun cuando las criptomonedas tienen un mundo completo por investigar y comentar, para la finalidad del artículo solamente será necesario conocer datos básicos y reconocer algunos de los factores incidentes en los precios de éstas; con dicha información se puede profundizar en torno a las arquitecturas para el análisis de grandes cantidades de datos en tiempo real, aplicadas a criptomonedas, en las que se pueda conocer cuál es el estado del arte, cuáles son los orígenes de información, los componentes principales, así como pros y contras, y ver qué relación existe entre estas arquitecturas y las arquitecturas de referencia como Lambda [1] y Kappa [3].

A partir de lo anterior, en el capítulo 2 de este artículo se presenta la metodología utilizada para la revisión hecha; en el marco teórico del capítulo 3 se habla de los conceptos más importantes para el entendimiento del artículo, en tanto que en el capítulo 4 se muestra el estado del arte de la aplicación de arquitecturas de procesamiento de datos en tiempo real a los sistemas de criptomonedas, y a la vez se conocerán diversas opiniones de algunos autores en dicho campo; en el capítulo 5 se analizarán los componentes utilizados por los autores para la generación de sus arquitecturas y se hará un estado comparativo entre las arquitecturas propuestas y las arquitecturas de referencia (kappa y lambda, según corresponda), y finalmente, en el último capítulo se verán las conclusiones y trabajos futuros.

METODOLOGÍA

Para el desarrollo de este artículo se han buscado bibliografías en sitios como IEEE, Google Scholar, Sciendo y Researchgate, utilizando palabras claves como *lambda architecture*, *kappa architecture*, *lambda implementation*, *real-time architecture*, *big data architecture*, *lambda cryptocurrency, kappa cryptocurrency, cryptocurrency real time architecture*, *cryptocurrency real time architecture analysis, cryptocurrency big data* y *cryptocurrency real-time*.

Así mismo, se consultaron páginas web como las de IBM o Oracle, para obtener información del marco teórico. Luego de la consecución de artículos, se hace la validación del estado del arte actual, para poder sacar conclusiones, beneficios, desventajas y comparaciones contra las arquitecturas de referencia.

MARCO TEÓRICO

A continuación, se presentan algunos conceptos teóricos claves para el entendimiento del artículo.

Big data

Big data es un término muy acuñado durante los últimos años, para el cual se encuentran múltiples definiciones, entre las cuales se destacan las siguientes:

Una de las más utilizadas nace en el año 2001, en el informe de Meta (ahora de Gartner) [9]; aun cuando en dicho informe no se menciona la palabra *big data*, se ha adoptado desde entonces como una de las definiciones más importantes.

Gartner propuso una definición en la que se emplean las tres V: volumen, velocidad y variedad. En el informe se destaca el tamaño cada vez mayor de los datos, lo que se denomina volumen; a su turno, la velocidad a la que se generan los datos es cada vez mayor, gracias a la cantidad de dispositivos capaces de producir datos, y por último la variedad, es decir, los datos, se generan en muchos formatos y representaciones diferentes [30].

Para IBM en [10], hace referencia al término *big data* como aquellas prácticas utilizadas para describir enormes cantidades de datos, teniendo en cuenta que éstos pueden ser estructurados, no estructurados y semiestructurados, y además tomaría demasiado tiempo y sería muy costoso cargarlos a una base de datos relacional para su análisis.

Sin embargo, el término *big data* no se refiere a una cantidad de datos en específico, pero sí se utiliza por lo general cuando se habla en términos de *petabytes* y *exabytes* de información. Al igual que la definición dada por Gartner [9], IBM y Microsoft [11] utilizan las tres V para la descripción de *big data*.

Por su parte, Oracle en [12] introduce dos nuevas V; ahora no solo son volumen, velocidad y variedad, sino también valor y veracidad.

Procesamientos de datos

El procesamiento de datos es la fase en la que se recogen, recopilan, limpian, filtran, agregan, transforman o enriquecen los datos para su posterior utilización. El procesamiento se inicia con datos en bruto, que se convierten a un formato más legible, dándoles el sentido y el contexto necesarios para su explotación. Existen dos paradigmas para llevar a cabo este proceso: *batch* y *streaming*.

- **Procesamientos tipo *batch***

Para entenderlo en el concepto *big data*, el procesamiento *batch* según [8] es una colección de datos que se han agrupado dentro de un intervalo de tiempo para ser procesados. Esto significa que la información no estará disponible en tiempo real, ya que se deberá esperar a que el intervalo de tiempo que se ha preestablecido se complete para que la información se pueda emplear.

Esto es muy común en el ámbito *big data*, para el manejo de información que no se requiere tener de manera inmediata. Si se consultan ejemplos de la vida real, una entidad financiera puede procesar todas las transacciones que ha hecho en un día; en este caso, el intervalo de tiempo específico será el citado anteriormente. Dichos datos contienen millones de registros, que se pueden almacenar como un archivo, el cual se procesa al final del día para varios análisis que la empresa desee hacer.

En el ámbito de las criptomonedas, el procesamiento *batch* puede servir para consolidar todas las transacciones que se realicen en un día de las criptomonedas o procesar precios históricos de la moneda para la predicción de precios y procesarlas para la toma de decisiones.

- **Procesamientos tipo *streaming***

Cada vez más, las personas y las empresas requieren tener acceso a los datos en el momento en que se generan, para poder tomar decisiones con base en información actualizada. Según [13], el procesamiento de *stream* permite procesar datos en tiempo real a medida que llegan y así se pueden detectar rápidamente las condiciones en un periodo de tiempo reducido desde el punto de recepción de los datos. Esta es la gran diferencia con los procesos *batch*, pues acá el tiempo de espera es muy reducido y a partir de ello se pueden procesar los datos a medida que van llegando; esto permite introducir datos en herramientas de análisis tan pronto como se generan y obtener resultados de análisis instantáneos. En el análisis del presente artículo, el procesamiento *streaming* será fundamental para procesar los datos a medida que se vayan generando, datos de redes sociales, de transacciones y de la oferta y la demanda de criptomonedas, lo que permitirá tomar las decisiones con base en la información recibida en tiempo real.

Arquitecturas big data para el procesamiento de datos en tiempo real

Antes de continuar, debemos aclarar qué es arquitectura y cómo vamos a utilizar este término dentro del artículo. La arquitectura se define como las estructuras, patrones, lineamientos o abstracciones de un sistema y las relaciones que existen entre ellos, hablando a un alto nivel [28].

La palabra *arquitectura*, aplicada en *big data* para el procesamiento de datos en tiempo real, se puede definir como aquellas abstracciones y relaciones necesarias para soportar grandes cantidades de información que llegan en tiempo real, es decir, en el momento en que se genera la información. En lugares como [7] y [14], se habla de las dos arquitecturas de referencia que cumplen con las dos condiciones planteadas (*big data* y *real-time*); estas arquitecturas son lambda y kappa.

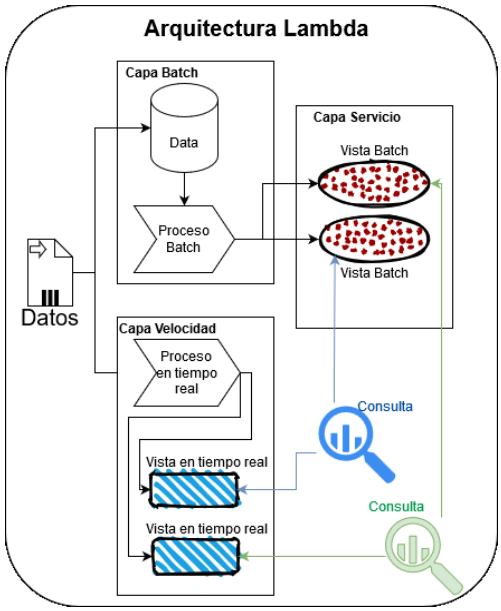

Figura 1. Diagrama de la arquitectura Lambda [1].

• Arquitectura Lambda

N. Marz [1, 29] propuso la arquitectura Lambda en el año 2011. Tenía como prioridad montar una arquitectura tolerante a fallos, ya sean de tipo humano o de *hardware*, y que cumpliera algunos objetivos que hoy en día requieren las arquitecturas *big data*, esto es, que sean escalables para que puedan crecer a medida que se requiera, y sobre todo con una baja latencia para escrituras y lecturas.

La arquitectura Lambda se divide en tres grandes capas: *batch*, servicio y velocidad. A renglón seguido se presentan los componentes y su interconexión (figura 1).

Ahora vamos a explicar un poco el funcionamiento de la arquitectura. Los datos que entran al sistema se envían tanto a la capa *batch* como a la capa de velocidad. La capa *batch* es la encargada de escribir los datos en el almacenamiento y luego los datos tendrán un procesamiento necesario para pasar a la capa de servicio. Esta última se encarga de indexar las vistas *batch*, de modo que pueda responder a las búsquedas o consultas con muy baja latencia y con los datos ya procesados; así, cuando se requiera acceder a los datos, no habrá necesidad de procesar todo el conjunto de datos, sino simplemente acceder a la vista.

El problema es que el proceso de escribir datos y luego indexarlos es bastante lento, por lo que éstos no están disponibles de manera instantánea [2].

Para ello, Marz propone la capa de velocidad, la cual se dedica a exponer sólo los datos más recientes, sin necesidad de preocuparse por escribirlos de manera permanente como sí sucede en la capa *batch* [16].

Al combinar estos dos paradigmas de procesamiento *batch* y procesamiento en tiempo real, cualquier búsqueda o consulta puede tomar datos provenientes tanto de vistas *batch* de la capa de servicio, como de las vistas en tiempo real de la capa de velocidad; esto ofrece lo mejor de dos mundos, ya que brinda un alcance completo y confiable por la capa *batch*, mientras que el modo *stream* nos da los datos en línea para decisiones instantáneas [2, 15].

• Arquitectura Kappa

J. Kreps [3] propuso la arquitectura Kappa en el año 2014. En esta propuesta critica la arquitectura Lambda debido al consumo innecesario de recursos que conlleva mantener y tratar los mismos datos, con el objetivo de obtener resultados similares, en dos sistemas distribuidos en la capa *batch* y en la capa velocidad.

Hay que entender que los procesamientos *batch* y los procesamientos *stream* manejan diversas herramientas; por lo tanto, el código está enfocado en cada una de ellas. Kreps afirma que el procesamiento *batch* es un subconjunto de las operaciones de *streaming*, y como

consecuencia, en su idea de Kappa plantea suprimir la capa *batch*, quedándose únicamente con la de velocidad y la de servicio, y pasando a considerar todo como un flujo de datos ininterrumpido.

A continuación, se presentan el diagrama de la arquitectura Kappa y el flujo que sigue en la actualidad (figura 2).

El funcionamiento es muy sencillo: la capa de tiempo real es la encargada de recibir el flujo de información, y de ahí se pasa a la capa de servicio, encargada de crear las vistas en tiempo real, con el objeto de que esté disponible para las consultas; esto simplifica el trabajo realizado en la arquitectura lambda.

La idea planteada se basa, primero que todo, en no modificar los datos de entrada, con el propósito de que si se requiere hacer un reproceso se cuente con ellos, tal cual se necesita. Al tener únicamente un flujo de procesamiento (*stream*), el manejo y el mantenimiento del procesamiento serán mucho más sencillos. Eso sí, los datos se deberán almacenar en una herramienta tipo Kafka, es decir, una herramienta que permita retener el registro completo de los datos de entrada y aún más si este tiempo de retención es personalizable, lo que permite hacer los reprocesos cuando sea necesario. Con esto, la arquitectura Kappa busca simplificar a la hora de construir las plataformas para el análisis de datos en tiempo real.

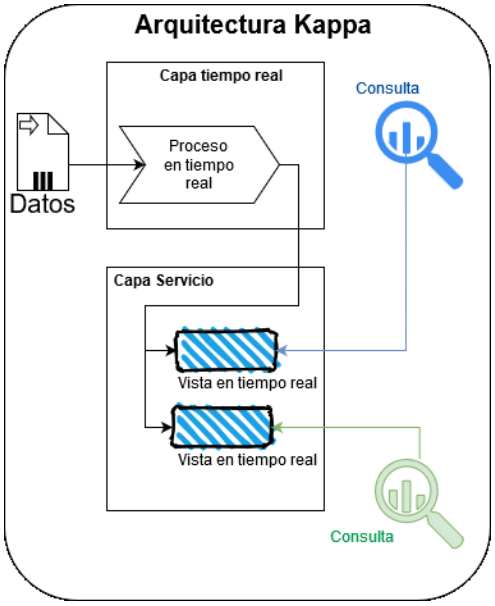

Figura 2. Diagrama de la arquitectura Kappa [3].

Criptomonedas

Una criptomoneda es un sistema de intercambio de *tokens* entre usuarios, respaldado y matemáticamente verificable en virtud de los mismos principios criptográficos que subyacen al cifrado en internet. Las criptomonedas se implementan típicamente como sistemas distribuidos (*peer-to-peer*), basados en las mismas tecnologías *blockchain* y que tienen el potencial de revolucionar los sistemas monetarios, financieros y de pago [19, 22].

Desde el año 2009, la economía mundial ha venido adoptando las criptomonedas como una parte de la economía; aunque todavía falta mucho para llegar a ser una realidad, poco a poco esta tecnología irá ganando adeptos.

Con *Bitcóin white*, de Satoshi Nakamoto [17], y el inicio de la red *peer-to-peer* de bitcóin, las criptomonedas han surgido como fenómenos tanto tecnológicos como económicos, atrayendo inversiones valoradas en billones de dólares a escala global. La tecnología *blockchain* tiene generalmente características claves de descentralización, persistencia, anonimato y auditabilidad [18].

En [22] se habla sobre el origen de las criptomonedas, las cuales nacen por la crisis de liquidez en los mercados financieros del 2009, cuando se publica un artículo de nueve páginas en un foro de internet titulado "Bitcóin: un sistema de dinero electrónico de igual a igual". Este artículo sería el empujón inicial para la creación de la primera criptomoneda, el bitcóin, basada en *blockchain*. En [33] se puede evidenciar la larga relación entre *blockchain* y *big data*.

ESTADO DEL ARTE

Horvat [6] ha utilizado la arquitectura Lambda como su eje central, enfocada en el procesamiento de datos en tiempo real de criptomonedas, teniendo en cuenta los siguientes factores: monitoreo en tiempo real de eventos de *blockchain*, estadísticas de minería, tendencias de compra y venta de criptomonedas, así como eventos de redes sociales relacionados con la reputación de criptomonedas.

Estos factores generan, por un lado, una necesidad de procesar datos históricos por el tema de estadísticas de minería, motivo por el cual se requieren ejecuciones *batch*, y por otro lado, la urgencia de procesar datos en tiempo real de los eventos de *blockchain*, redes sociales y tendencias; esta necesidad hace que la arquitectura

que mejor se acople sea la Lambda, para tener menor latencia, por sus capas *batch*, velocidad y servicio.

A renglón seguido (figura 3) se puede observar la arquitectura propuesta por Horvat, que tiene una gran similitud con la arquitectura Lambda (figura 1): las mismas capas *batch*, velocidad y servicio.

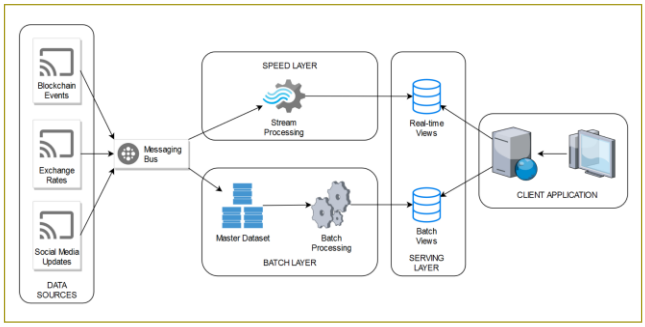

Figura 3. Arquitectura propuesta en [6].

El primer componente propuesto en esta arquitectura es la fuente de datos, conformada por las transacciones pendientes enviadas recientemente, los nuevos bloques agregados al *blockchain,* los datos de intercambio de criptomoneda en tiempo real relacionados con el precio de ésta, las menciones de la criptomoneda en las redes sociales como Twitter y Reddit, y datos no relacionados directamente con la criptomoneda, pero que pueden afectar datos de la bolsa global y cambios en el precio del petróleo.

Como una pequeña variación o paso intermedio a la arquitectura lambda, se ha decidido utilizar un bus de mensajería para recibir la información. Se emplea Kafka como plataforma de mensajería distribuida, tolerante a fallas y rápida mediante técnicas de partición y paralelización; ésta será la responsable de recibir la información que llega y transmitirla a las capas *batch* y de velocidad.

La información en la capa *batch* se almacenará en HDFS, el sistema de archivos de Hadoop, ya que es distribuido, escalable, portátil y tolerante a fallas; para procesar los datos dentro de la misma capa se usa Spark.

Por el camino de *real-time*, los datos llegan a la capa de velocidad bajo Spark Streaming o Kafka Streams, que permite una capacidad de procesamiento escalable y de alto rendimiento en la transmisión de datos; ahí también se hacen procesamientos como filtrado, mapeo, agregación, cruces y uniones.

Y por último, la capa de servicio, en la que se ha decidido utilizar Apache Druid (sistema de administración de bases de datos orientado a columnas y distribuido de código abierto, que combina ideas de bases de datos analíticas, bases de datos de series de tiempo y sistemas de búsqueda para habilitar casos de uso en arquitecturas de transmisión) [6], como base de datos para almacenar los registros de datos requeridos de las vistas *batch* y en tiempo real. Esta capa de servicio debe garantizar una rápida consulta de los datos, con el beneficio de poder mezclar tanto las consultas *batch* como las consultas en tiempo real.

Se ha detallado la implementación de algunas herramientas, para plasmar la arquitectura lambda en el problema; cabe aclarar que estas herramientas nombradas no son ni las mejores, ni las más utilizadas, ni las únicas. Esta referencia encontrada decidió emplearlas y cumple a cabalidad con lo requerido en la arquitectura.

En [19] utilizan redes sociales e información de la Deep Web como fuente de datos; su arquitectura se centra en una herramienta para el procesamiento de lenguaje natural (PLN), a partir de los orígenes de información. Los datos recibidos pasan por GATE (*General Architecture for Text Engineering*), una plataforma para la creación rápida de prototipos de aplicaciones de PLN, la cual será la encargada de mostrar los resultados a dichos análisis.

Lo mismo sucede con [20], que utiliza el análisis de sentimientos en las redes sociales y determina la correlación entre ésta y las criptomonedas.

En [21] se puede encontrar un tema un poco distinto, pero muy relacionado con el caso de uso; dentro de su arquitectura se define un componente de análisis predictivo, puesto que se emplean redes neuronales convolucionales. Sus orígenes de datos son el precio histórico de un conjunto de activos financieros y precios de criptomonedas.

En [23] se plantea una plataforma de predicción de precios de criptomonedas en tiempo real y adaptativa, basada en los sentimientos de Twitter. La arquitectura propuesta se fundamenta en tres puntos principales (figura 4). El primero es una capa basada en Spark, que maneja el gran volumen de datos entrantes de manera persistente y tolerante a fallas; el segundo es un enfoque que respalda el análisis de sentimientos que puede responder a grandes cantidades de consultas de procesamiento del lenguaje natural en tiempo real, y el tercero es un método predictivo, fundado en el aprendizaje en línea. Se plantean dos grandes flujos dentro

de la arquitectura; por un lado, se tiene la parte de datos históricos en la parte inferior de la figura siguiente (figura 4), los cuales pasan por un procesamiento de datos, encargado de enriquecer la información; posteriormente llegan al componente del modelo predictivo, para finalizar en la capa de actualización del modelo de datos en tiempo real.

Por otro lado, encontraremos el flujo de información de tiempo real (Twitter + *currency data stream*), pasa por Vader y llega a la parte de actualización del modelo de datos en tiempo real, donde se encuentra con los datos históricos, para posteriormente estar disponibles en la capa de visualización.

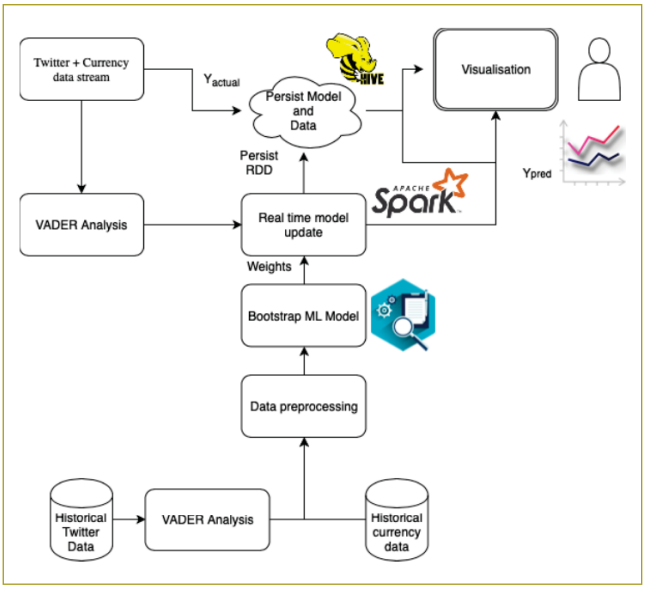

Figura 4. Arquitectura de KriptoOracle [23].

En el artículo [24] de Ayvaz se tomó como caso de estudio la relación entre las opiniones públicas en las redes sociales sobre las criptomonedas y los cambios en sus precios, utilizando enfoques de análisis de sentimiento basados en el léxico, con el objetivo de evaluar la viabilidad de predecir los precios de las criptomonedas.

Para esto proponen una clase de arquitectura (figura 5), y se determina como origen de información la red social Twitter, una capa de *streaming* (*data streaming layer*), que funcionará con la herramienta Spark Streaming, la cual permite el procesamiento de datos escalable, de alto rendimiento y tolerante a fallas en tiempo real, así como una capa de procesamiento (*data processing layer*).

Acá se utilizarán dos herramientas. Por una parte, se tiene Hive, que es un *software* de almacenamiento de datos que simplifica la escritura, lectura y gestión de grandes conjuntos de datos en almacenamientos distribuidos, y por otra parte está Apache Spark, que es un marco computacional distribuido para el procesamiento de grandes datos; hay una capa de almacenamiento (*data storage layer*) que utilizará la herramienta HDFS (*Hadoop Distributed File System*), la cual es un sistema de archivos distribuido, altamente tolerante a fallas, que maneja grandes conjuntos de datos, y otra capa de visualización (*view layer*), que será la encargada de mostrar los resultados de los procesamientos; para esto han planteado la utilización de la herramienta Zeppelin, que es un *notebook* de código abierto basado en la web que permite la ingesta, el descubrimiento, el análisis y la visualización de la analítica de datos.

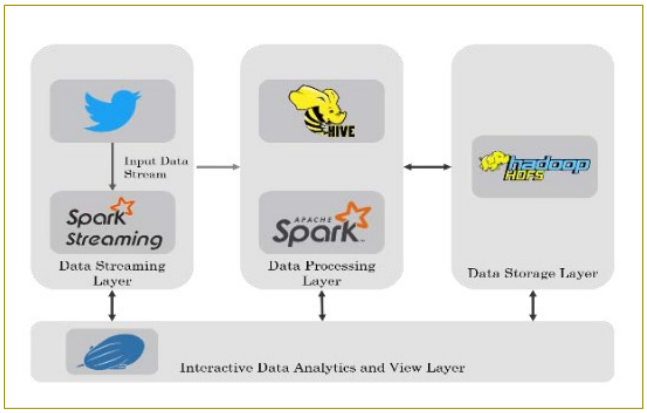

Figura 5. Arquitectura propuesta en [24].

En artículos como [31, 32] se busca consolidar información entre *big data* y criptomonedas o *blockchain*; aunque no hablan de arquitecturas específicas, ayudan a entenderlas, puesto que presentan soluciones novedosas asociadas con algunas de las áreas de *big data* que se pueden potenciar por medio de la tecnología *blockchain*, realizando una revisión sistemática de las interacciones entre *big data* y criptomonedas, y servir como un directorio de referencia.

Si bien en el artículo se pretende revisar arquitecturas *big data* para el análisis de datos en tiempo real en el contexto de criptomonedas, se quiere dejar en evidencia que éste no es el único ámbito en el que se pueden aplicar tales arquitecturas, ya que existen otros tipos de aplicaciones; por ejemplo, en [4] se emplea en el contexto de restaurantes, y en [34] el ámbito de aplicabilidad es con sensores IoT, lo que abre muchos

caminos, como en ciudades inteligentes [36], temas de salud con dispositivos médicos [37] y temas de *marketing* [35], pero todavía hay muchos ámbitos en los que se pueden utilizar.

ANÁLISIS COMPARATIVO

Encontrar un estándar para dicho problema no es nada sencillo, debido a todas las variantes que se puedan hallar en la definición de la solución, desde los orígenes de los datos, hasta las propuestas de las arquitecturas y las herramientas seleccionadas. Eso sí, las principales características que deben cumplir las arquitecturas es que deben trabajar con grandes cantidades de datos, en tiempo real y aplicadas a criptomonedas, para que la toma de decisión sea lo más cercana a la actualidad y esto, a su vez, lleva a adoptar mejores decisiones por tener la información más reciente.

Para un mejor entendimiento, se propone una categorización para el estado del arte; las arquitecturas se podrán categorizar en *real-time*, *batch*, analítica, procesamiento de lenguaje natural y seguridad.

En su artículo [6], Horvat toma como referencia la arquitectura lambda para la construcción de su solución; da una explicación clara de las herramientas que utiliza en cada una de las capas propuestas por Marz, pero hace una pequeña variación al introducir un bus de mensajería. De resto, cada capa tiene relación con una herramienta específica que cumple a cabalidad la funcionalidad propuesta en la arquitectura de referencia.

Al analizar las ventajas de la arquitectura utilizada se puede encontrar la baja latencia, gracias a la combinación del mundo *batch* y el *stream*. Por un lado, en la capa *speed* se tiene la información en tiempo real, fundamental para una arquitectura enfocada en criptomonedas, y por el otro lado, en la capa *batch* la información histórica, que por su volumen es más pesada para procesar y en herramientas *real-time* no es nada óptimo, pero eso sí, de suma importancia para la toma de decisiones en el ámbito de las criptomonedas, de acuerdo con los principales factores incidentes en el valor de las criptomonedas [5].

Así mismo, las desventajas o críticas a la arquitectura propuesta tienen que ver con el problema de administrar y mantener dos sistemas distribuidos complejos: *batch* y *streaming*. Esto lo hace estructuralmente complejo. La integración de tantas herramientas puede generar un hueco en temas de seguridad, pero los autores no se preocupan por el manejo de este tema; además, falta un componente dentro de la arquitectura que permita un manejo de análisis predictivo, fundamental en el campo de las criptomonedas. Esta arquitectura entra en las categorías *real-time* y *batch*.

En [19] existe una ventaja, pues dentro de su arquitectura posee una herramienta para el procesamiento de lenguajes naturales, pero realmente no cumple con los principios requeridos, que sea capaz de manejar grandes cantidades de información en tiempo real. Así mismo, compararlo con alguna de las arquitecturas de referencia no tiene sentido, no encaja en ninguna de las dos, le faltan más orígenes de información para que los resultados se acerquen más a lo esperado en criptomonedas, no hay componentes de seguridad ni componentes predictivos; ésta entra en la categoría de procesamiento de lenguaje natural y *batch*.

Para [21] dentro de su arquitectura definen un componente de análisis predictivo, pues se utilizan redes neuronales convolucionales, sus orígenes de datos son el precio histórico de un conjunto de activos financieros y precios de criptomonedas. Dichos modelos pueden emplearse para predicciones usando datos históricos, pero carecen de una forma conveniente y eficiente de procesar nuevos datos en tiempo real. Se categoriza en *batch*, analítica.

Mohapatra en [23] no utiliza ninguna de las arquitecturas de referencia mencionadas en este artículo, ni lambda ni kappa; por el contrario, ha creado una arquitectura propia. Si queremos hacer una comparación entre las arquitecturas de referencia, podemos evidenciar que Mohapatra habla implícitamente de la arquitectura lambda y plantea dos flujos dentro de la arquitectura: el flujo de datos históricos, asociado a una capa *batch*, y un flujo de datos en tiempo real, asociado a la capa de velocidad de Marz.

Por el lado de datos históricos maneja el procesamiento de información, encargado de funciones como filtrado, agregación, uniones y cruces; esta información pasa por un componente clave de los modelos predictivos, tan necesarios para el análisis de las criptomonedas. Por último, tiene un componente de visualización asociado a la capa de servicio en la arquitectura lambda.

Esta arquitectura maneja componentes históricos y en tiempo real, lo que ayuda a que la información obtenida sea más precisa y, sobre todo, más rápida de consultar; además, hace hincapié en algo a lo que ningu-

na otra referencia le ha dado la suficiente importancia: el componente predictivo dentro de la arquitectura.

Al igual que en las otras arquitecturas, falta la seguridad dentro de ésta, que se puede categorizar en *batch*, *real-time* y analítica.

Ayvaz en [24], aun cuando no nombra la arquitectura Lambda como su referencia, hace un símil entre la arquitectura de referencia y la propuesta, y se evidencia que tiene una gran similitud. La capa de *data streaming layer* cumple la misma funcionalidad de la capa de velocidad de Lambda, esto es, se procesa toda la información que va llegando en tiempo real.

Una de las diferencias es la capa *data storage layer*. Marz no propone una capa específica para esta acción, sino que afirma que dentro de la capa *batch* se encuentra este almacenamiento de los datos. Por lo tanto, si se ve como una sola capa, la similitud con esta arquitectura de referencia es aún mayor.

La capa *data processing layer* cumple la misma función de la capa *batch*; en ésta se procesa toda la información necesaria para poder hacer todo el análisis del caso de uso propuesto. Por otro lado, la *view layer* se compara con la capa de servicio de Lambda, puesto que será la encargada de visualizar la información gracias al procesamiento *batch* y al procesamiento *stream*.

La simplicidad es una de las ventajas de la arquitectura propuesta. Se tiene una capa para el procesamiento *batch*, otra para *stream*, otra de almacenamiento y otra de visualización. Una desventaja es la falta de componente predictivo, de seguridad; una capa exclusiva para el almacenamiento es muy grande, por lo que debe estar intrínseco en la capa *batch*. Se categoriza en *real-time* y *batch*.

Este análisis comparativo, sumado al estado del arte, lleva a evidenciar la falta de bibliografías en la que se utilice la arquitectura Kappa como referencia para las arquitecturas *big data* para el análisis de datos aplicado a criptomonedas.

Pese a que se supone que Kappa es una simplificación de lambda, no siempre constituye una solución viable para todos los casos, y este es uno de ellos; depende de factores como el tipo de procesamientos simultáneos de los datos que se pretenden realizar. Cuando se analizan criptomonedas, hay que validar tanto datos históricos como datos recientes; esto hace que la arquitectura Lambda sea la predilecta, pues combina los paradigmas *batch* y *stream*.

Por otro lado, la importancia de la latencia para los actores implicados es fundamental, especialmente en un ámbito como el de las criptomonedas, en el que se requiere tener la información demasiado rápido para la toma de decisiones.

CONCLUSIONES Y TRABAJO FUTURO

Los resultados del procesamiento de datos en tiempo real pueden producir una ganancia económica potencial al predecir las fluctuaciones de precios, a causa de la naturaleza volátil de las criptomonedas; esto motiva a encontrar la arquitectura más adecuada para procesar cantidades tan grandes de datos, sobre todo en tiempo real, debido a la necesidad que tienen los expertos de monitorear y analizar algunos tipos de eventos relacionados con el mercado de estas monedas, para la toma de decisiones fundadas en información en tiempo real.

Las arquitecturas de referencia lambda y kappa tienen sus pros y sus contras, pero lo que sí hay que destacar es que ambas satisfacen la necesidad del análisis de datos en tiempo real. Para el caso de las criptomonedas, la arquitectura más acorde es lambda, por lo siguiente: si analizamos cuáles son los principales factores que afectan los precios de las criptomonedas tenemos, por un lado, la oferta y la demanda. Sobre esto necesitamos observar transacciones históricas y en tiempo real, los sentimientos de pánico, pesimismo, escepticismo, optimismo y euforia, así como la opinión de las personas; será fundamental conocerlos en tiempo real.

Por otro lado, tenemos el análisis del mercado, que también es un factor determinante, puesto que servirá para evaluar el precio actual y el precio histórico de las criptomonedas; todos estos atributos conducen a la necesidad de que se maneje el procesamiento *batch* para temas históricos, ya que se requiere procesar grandes cantidades de datos y procesamiento *stream*, al igual que procesar la información en tiempo real. Lambda utiliza estos dos paradigmas y por eso la mayoría de las referencias se inspiran en esta arquitectura.

La investigación hecha dejó como resultado que existe una gran falencia a la hora de trabajar las arquitecturas *big data* aplicadas a criptomonedas en un componente fundamental: la seguridad. Ninguno de los autores se preocupa por tratar la seguridad dentro de las arquitecturas, como si éste no fuera un tema clave; hay que reconocer que la seguridad debe ser un tema transversal

a toda la arquitectura, que debe estar presente en cada una de las capas y entre sus interconexiones.

Otro asunto muy asociado es la privacidad de la información, aspecto que los autores tampoco trataron, razón por la cual resulta indispensable validar este tema como trabajo futuro y evaluar hoy en día qué tendencias se están tratando en esta clase de arquitecturas para controlar la seguridad.

Así mismo, las arquitecturas *big data* y las criptomonedas requieren un componente importante que también se le resta importancia en las arquitecturas descritas en el estado del arte: la capa predictiva, estas arquitecturas deberían utilizar más la inteligencia artificial [31], como uno de los componentes principales, temas como *machine learning*, *deep learning* y análisis predictivo deberán cobrar más importancia en los planteamientos futuros.

Otra de las falencias evidenciadas tiene que ver con la veracidad de las fuentes de información, que si bien no están completamente relacionadas con las arquitecturas, es importante nombrar el tema, puesto que forma parte del flujo en toda la arquitectura, y más cuando estas arquitecturas, aplicadas al contexto de criptomonedas, requieren información como redes sociales para el análisis de sentimientos. Por todo esto, tales arquitecturas deben preocuparse un poco más respecto a los orígenes de la información.

Otro punto para un trabajo futuro es el referente al despliegue de las arquitecturas, en especial lo relacionado con hallar el mejor lugar para desplegarlas, debido a la variedad de opciones que hoy en día se presentan.

En ese orden de ideas, la computación en la nube nos abre muchas puertas para llegar a implementar dichas arquitecturas y poder ir escalando tanto horizontal como verticalmente, además de todo lo que nos puede ofrecer; por ejemplo, infraestructura como servicio, plataforma como servicio y *software* como servicio, sin mayor desgaste. Así mismo, hay que evaluar cuál será la mejor opción, si una arquitectura en nube híbrida, en nube pública, en nube privada u *on premise*.

En virtud de todo ese análisis, se ha decidido proponer una arquitectura en la que se puedan evidenciar todas las capas y componentes requeridos para el análisis de datos en tiempo real, aplicada a criptomonedas.

La figura siguiente (figura 6) se basa en la arquitectura Lambda como fuente de inspiración, incluyendo dos componentes adicionales. Por un lado, la capa predictiva necesaria para todos los análisis, en la que se podrán hacer modelos predictivos, aprendizaje automático, análisis de sentimientos, entre otros; es un componente muy potente para este tipo de arquitecturas. Y, por otro lado, está la capa de seguridad, transversal a toda la arquitectura. De resto se utilizará la capa *batch*, la capa *streaming* o velocidad y la capa de servicio que propone N. Marz. Esta arquitectura cumpliría a cabalidad con lo requerido en este contexto de criptomonedas.

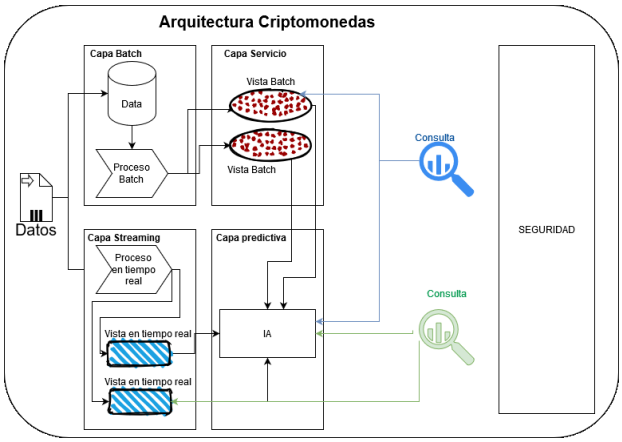

Figura 6. Arquitectura de criptomonedas propuesta.

Las arquitecturas lambda y kappa no solo se pueden utilizar para arquitecturas relacionadas con criptomonedas. En cualquier trabajo que requiera análisis de grandes cantidades de datos en tiempo real se puede elegir alguna de éstas para implementar o para basarse como arquitecturas de referencia.

El presente artículo es producto del trabajo del autor como estudiante de la Maestría en Informática, en el desarrollo de la asignatura Nuevas Tecnologías de la Información, a cargo de la ingeniera Claudia Patricia Santiago Cely.

REFERENCIAS

[1] Marz, N. (2011, octubre). How to beat the CAP theorem [en línea]. http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html [últimoacceso: 01/11/2020].
[2] Feick, M., Kleer, N. & Kohn, M. (2018). Fundamentals of real-time data processing architectures lambda and kappa. Lecture Notes in Informatics (LNI), 1.
[3] Kreps, J. (2014, julio). Questioning the lambda architecture. O'Reilly [en línea]. https://www.oreilly.com/radar/questioning-the-lambda-architecture/ [último acceso: 21/11/2020].

[4] Lee, C.-H. & Lin, C.-Y. (2017). Implementation of lambda architecture: a restaurant recommender system over Apache Mesos. IEEE 31st International Conference on Advanced Information Networking and Applications (AINA). Taipéi.
[5] Godoy, G. (2020). ¿Qué es lo que mueve el precio del bitcóin? ¿Cuál es el factor más importante? [en línea]. https://es.cointelegraph.com/news/what-drives-the-price-of-bitcóin-whats-the-most-important-factor [último acceso: 30/ 11/2020].
[6] Horvat, N., Ivković, V., Todorović, N., Ivančević, V., Gajic, D. & Luković, I. (2020). Big data architecture for cryptocurrency real-time data processing. ICIST 2020 - 10th International Conference on Information Society and Technology. Kopaonik, 2020.
[7] Verrilli, M. (2017, agosto). De lambda a kappa: guía sobre arquitecturas de big data en tiempo real. Talend [en línea]. https://www.talend.com/es/blog/2017/08/28/lambda-kappa-real-time-big-data-architectures/ [último acceso: 21/11/2020].
[8] Balkenende, M. (2018, junio). The big data debate: batch versus stream processing. The New Stack [en línea]. https://thenewstack.io/the-big-data-debate-batch-processing-vs-streaming-processing/ [último acceso: 21/11/2020].
[9] Laney, D. (2001). 3D Data management: controlling data volume, velocity, and variety. META Group.
[10] Barranco Fragoso, R. (2012, junio). ¿Qué es big data? IBM [en línea]. https://developer.ibm.com/es/articles/que-es-big-data/ [último acceso: 21/11/2020].
[11] Microsoft (2015). ¿Qué es eso llamado big data? 24/08/2015 [en línea]. https://news.microsoft.com/es-xl/que-es-eso-llamado-big-data/ [último acceso: 23/ /11/2020].
[12] Oracle (2020). ¿Qué es big data? [en línea]. https://www.oracle.com/co/big-data/what-is-big-data.html [último acceso: 23/11/2020].
[13] Vaseekaran, G. (2017). Big data battle: batch processing vs. stream processing. 21/10/2017 [en línea]. https://medium.com/@gowthamy/big-data-battle-batch-processing-vs-stream-processing-5d94600d8103 [último acceso: 23/11/2020].
[14] Domínguez, J. (2018). De lambda a kappa: evolución de las arquitecturas big data [en línea]. https://www.paradigmadigital.com/techbiz/de-lambda-a-kappa-evolucion-de-las-arquitecturas-big-data/ [último acceso: 23/11/2020].
[15] Vera-Tudela, B. (s.f.). Arquitectura lambda: combinando lo mejor de dos mundos. SG [en línea]. https://sg.com.mx/revista/52/arquitectura-lambda-combinando-lo-mejor-dos-mundos [último acceso: 21/11/2020].
[16] Bryan, P. (2018, noviembre). Big data y la arquitectura lambda [en línea]. https://medium.com/big-data-world/big-data-y-la-arquitectura-lambda-f571e117670a [último acceso: 23/11/2020].
[17] Nakamoto, S. (2008). Bitcóin: a peer-to-peer electronic cash system.
[18] Zheng, Z., Xie, S., Dai, H., Chen, X. & Wang, H. (2017). An overview of blockchain technology: architecture, consensus, and future trends. 6th IEEE International Congress on Big Data. Honolulú.
[19] Laskowski, L. & Kim, H. (2016). Rapid prototyping of a text mining application for cryptocurrency market intelligence. 5th IEEE International Workshop on Data Integration and Mining (IEEE DIM). Pittsburgh.
[20] Dulau, T. & Dulau, M. (2019). Cryptocurrency: sentiment analysis in social media. Acta Marisiensis. Serie Technologica, XVI (2), 1-6.
[21] Jiang, Z. & Liang, J. (2017). Cryptocurrency portfolio management with deep reinforcement learning. 2017 Intelligent Systems Conference (IntelliSys). Londres.
[22] Ordinas, M. (2017). Las criptomonedas: ¿oportunidad o burbuja? Palma de Mallorca: Banca March.
[23] Mohapatra, S., Ahmed, N. & Alencar, P. (2019). KryptoOracle: a real-time cryptocurrency price prediction platform using twitter sentiments. 2019 IEEE International Conference on Big Data (Big Data). Los Ángeles.
[24] Ayvaz, S. & Shiha, M. (2018). A scalable streaming big data architecture for real-time sentiment analysis. Proceedings of the 2018 2nd International Conference on Cloud and Big Data Computing (ICCBDC'18). Nueva York.
[25] CoinMarketCap (2020, noviembre) [en línea]. https://coinmarketcap.com/es/ [último acceso: 30/11/2020].
[26] Reiff, N. (2020, enero). The 10 most important cryptocurrencies other than bitcóin [en línea]. https://www.investopedia.com/tech/most-important-cryptocurrencies-other-than-bitcóin/ [último acceso: 30/11/2020].
[27] Charts, Y. (2020, noviembre). Ethereum transactions per day [en línea]. https://ycharts.com/indicators/ethereum_transactions_per_day [último acceso: 30/ 11/2020].
[28] Bass, L., Clements, P. & Kazman, R. (2003). Arquitectura de software. Boston: Addison Wesley.
[29] Marz, N. & Warren, J. (2015). Big data: principles and best practices of scalable real time data systems. Nueva York: Manning Publications Co.
[30] Demchenko, Y., Laat, C. de & Membrey, M. (2014). Defining architecture components of the Big Data Ecosystem. International Conference on Collaboration Technologies and Systems (CTS). Minneápolis: IEE.
[31] Hassan, H., Huang, X. & Silva, E. (2018). Big-crypto: big data, blockchain and cryptocurrency. Big Data and Cognitive Computing, 2(4), 34.
[32] Karafiloski, E. & Mishev, A. (2017). Blockchain solutions for big data challenges: a literature review. IEEE Eurocon 2017-17th International Conference on Smart Technologies. Ohrid.
[33] Deepa, N., Pham, Q.-V.,. Nguyen, D. C., Bhattacharya, S., Boopathy, P., Gadekallu, T. R., Reddy, P. K., Fang, F. & Pathirana, P. N. (2020). A survey on blockchain for big data: approaches, opportunities, and future directions. ACM Comput. Surv. 1(1).
[34] Ullah Rathore, M. M., Paul, A., Ahmad, A., Chen, B.-W., Huang, B. & Ji, W. (2015). Real-time big data analytical architecture for remote sensing application. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8, 4610- 4621.
[35] Jabbar, A., Akhtarb, P. & Dani, S. (2020). Real-time big data processing for instantaneous marketing decisions: a problematization approach. Industrial Marketing Management, 90 (1), 558-569.
[36] Silva, B. N., Khan, M., Jung, C., Seo, J., Muhammad, D., Han, J., Yoon, Y. & Han, K. (2018). Urban planning and smart city decision management empowered by real-time data processing using big data analytics. Sensors, 18.
[37] Chrimes, D., Kuo, M.-H., Moa, B. & Hu, W. (2017). Towards a real-time big data analytics platform for health applications. International Journal of Big Data Intelligence, 4(2).

REVISTA DE LA ESCUELA COLOMBIANA DE INGENIERÍA

Alcance y política

El objetivo de la *Revista de la Escuela Colombiana de Ingeniería* es difundir artículos técnicos que contribuyan al desarrollo del país a través de una publicación con alta calidad editorial y rigor científico.

La revista acepta prioritariamente los siguientes tipos de trabajos, que le permiten mantener su categorización:

1. **Artículo de investigación científica y tecnológica**. Documento que presenta, de manera detallada, los resultados originales de proyectos de investigación. La estructura generalmente utilizada contiene cuatro apartes importantes: introducción, metodología, resultados y conclusiones.
2. **Artículo de reflexión**. Documento que presenta resultados de investigación desde una perspectiva analítica, interpretativa o crítica del autor, sobre un tema específico, recurriendo a fuentes originales.
3. **Artículo de revisión**. Documento producto de una investigación donde se analizan, sistematizan e integran los resultados de investigaciones publicadas o no publicadas, sobre un campo en ciencia o tecnología, con el fin de dar cuenta de los avances y las tendencias de desarrollo. Se caracteriza por presentar una cuidadosa revisión bibliográfica.

También admite artículos de las siguientes tipologías:

4. **Artículo corto**. Documento breve que presenta resultados originales preliminares o parciales de una investigación científica o tecnológica, que por lo general requieren una pronta difusión.
5. **Reporte de caso**. Documento que presenta los resultados de un estudio sobre una situación particular, con el fin de dar a conocer las experiencias técnicas y metodológicas consideradas en un caso específico.
6. **Revisión de tema**. Documento resultado de la revisión crítica de la bibliografía sobre un tema en particular.

Cabe destacar que se privilegian para la revista los tipos de artículos de los numerales 1, 2 y 3.

La revista circula trimestralmente y recibe sólo artículos inéditos. Los trabajos recibidos se someten al concepto de pares académicos y del Consejo Editorial.

Requisitos para la publicación de artículos

Los artículos presentados a la revista deben remitirse por correo electrónico a *revista@escuelaing.edu.co*, adjuntando los siguientes formatos debidamente diligenciados: autor.doc, clasificación.doc y tipo.doc, cuyos archivos se pueden descargar de http://www.escuelaing.edu.co/revista.htm. En este mismo sitio está disponible la plantilla guía que contiene la estructura determinada por la revista para los artículos.

Scope and policy

Revista de la Escuela Colombiana de Ingeniería disseminates technology articles helping to our country development. It emphasises on its high quality print and its scientific rigour. Articles submitted for publication shall be classified into one of the following categories– which allow it keeps its indexation:

1. **Scientific and technological research article**. These documents offer a detailed description about the original findings of research projects. In general, the usually used structure contains four important sections: introduction, methodology, results and conclusions.
2. **Reflection article**. These documents present the results of a research project on a specific, interpretative, or critical view by the author about a particular topic by using original sources.
3. **Review**. A document resulting from a finished research, where the published and/or unpublished findings of investigation in a particular field of science or technology are analysed, systematised and integrated to report the progress and the development tendencies. These documents include a careful bibliographic review.

Revista de la Escuela Colombiana de Ingeniería also accepts the following types of articles:

4. **Short article**. A brief text presenting the original, preliminary and/or partial results of a scientific or technological study, which normally need to be disseminated as quickly as possible.
5. **Case report**. A document that presents the results of a study on a specific situation in order to report the technical and methodological experiences considered in a particular case.
6. **Thematic review**. These documents are the product of a critical review of literature on a particular topic.

Our revista privilege articles as the highlight ones in numbers 1, 2 and 3.

Revista de la Escuela Colombiana de Ingeniería is a quarterly publication that only accepts unpublished articles. The revista submits all the papers to the verdict of two academic peers, who evaluate the article.

Ruling for publication

The article must be sent by e-mail to *revista@escuelaing.edu.co* with 3 files attached: Author.doc, Classification.doc and Type.doc available in http://www.escuelaing.edu.co/revista.htm. There is also a template guide for the structure of the article (template guide.doc).