Dossier. Analítica de datos

Analítica de grafos para identificar entidades relevantes y comunidades en Mercado Libre: un estudio de caso

Use of Graph Analytics to Identify Relevant Entities and Communities in Mercado Libre: A Case Study

Carrillo Gelvez, Gerson Enrique; Galpin, Ixent

Gerson Enrique Carrillo Gelvez
gersone.carrillog@utadeo.edu.co
Universidad de Bogotá Jorge Tadeo Lozano, Colombia
Ixent Galpin
ixent.galpin@utadeo.edu.co
Universidad de Bogotá Jorge Tadeo Lozano, Colombia

Revista Mutis
Universidad de Bogotá Jorge Tadeo Lozano, Colombia
ISSN: 2256-1498
Periodicidad: Semestral
vol. 11, núm. 1, 2021
revista.mutis@utadeo.edu.co

Recepción: 18 Enero 2021
Aprobación: 30 Marzo 2021

URL: http://portal.amelica.org/ameli/journal/193/1933515007/

DOI: https://doi.org/10.21789/22561498.1740

Resumen: Este artículo representa la información disponible en bases de datos no relacionales, aprovechando los beneficios de escalabilidad, alta disponibilidad, resiliencia y facilidad proporcionados por estas. Así mismo, se da a conocer una serie de algoritmos suministrados por el motor de bases de datos de grafos Neo4j para computar métricas de grafos, nodos y relaciones. En primer lugar, se consolida un conjunto de datos públicos tomado del sistema de ventas online de Mercado Libre. Posteriormente, se modelan los datos obtenidos en un esquema de grafos que tiene como nodos a los usuarios, quienes pueden ser vendedores, compradores, productos y sus características. Como siguiente paso, se aplican algoritmos que calculan métricas del grafo, junto con sus nodos y relaciones, visualizando de esta manera los resultados obtenidos. Para finalizar, se identifican las categorías ofertadas más importantes, las comunidades existentes y los usuarios más influyentes.

Palabras clave: base de datos de grafos, analítica de grafos, NoSQL, métricas de centralidad, detección de comunidades.

Abstract: This article represents the information available in non-relational databases, taking advantage of their scalability, high availability, resilience, and ease of development. This work also describes some algorithms provided by the Neo4j graph database engine to compute graph, node and relationship metrics. To do this, we first consolidate a data set obtained from Mercado Libre online sales system. Subsequently, the data is cast into a graph schema that considers users as nodes. Such users can be sellers or buyers, products and their characteristics. Afterward, we applied the algorithms that calculate metrics from the graph, as well as its nodes and relationships, thus displaying the results obtained. Finally, we identify the most important categories offered, along with the most influential communities and users.

Keywords: Graph database, graph analytics, NoSQL, centrality metrics, community detection.

INTRODUCCIÓN

Desde la década de 1970, las bases de datos relacionales han funcionado de manera exitosa con datos que encajan en el formato de tablas, columnas y filas. No obstante, las consultas sobre conjuntos de datos con entidades altamente interrelacionadas son complejas de expresar en SQL, puesto que a menudo requieren de un alto número de joins y tienden a consumir excesivos recursos de ejecución. De ahí el surgimiento de las bases de datos de grafos, las cuales están enfocadas hacia datos conectados y permiten a las organizaciones comprender la gran cantidad de conexiones existentes entre entidades (Svensson, 2020). El uso de las bases de datos NoSQL en proyectos de desarrollo de software ha ido incrementando (Pragma, 2018). Por este motivo, es importante explorar ejemplos de cómo se puede modelar una base de datos de grafos mediante la aplicación de algoritmos, teniendo en cuenta que los volúmenes de datos de los sistemas crecen cada día más y más.

DB-Engines (2020a) clasifica los sistemas de administración de bases de datos según su popularidad en un ranking que se actualiza mensualmente. Para el mes de octubre de 2020, las bases de datos relacionales ocuparon los 4 primeros puestos (Oracle, MySQL, Microsoft SQL Server y PostgreSQL), mientras que el quinto lugar fue ocupado por la base de datos de documentos MongoDB. El puesto número 21 del ranking fue ocupado por la base de datos orientada a grafos Neo4j, lo que indica que este tipo de bases viene ganando popularidad.

Cuando se crea un nuevo sistema de información se seleccionan motores de bases de datos diferentes a los orientados a grafos. Posiblemente, esto se debe a que se carece de conocimiento sobre las propiedades y los beneficios que este tipo de motores pueden traer a la solución. Este mismo desconocimiento imposibilita identificar que se puede contar con conjuntos de datos limitados para evaluar y probar las características de las bases de datos de grafos, dentro de las que Neo4j, Microsoft Azure Cosmos DB, ArangoDB, OrientDB y Virtuoso son las 5 más populares (DB-Engines, 2020b). A pesar de esto, los grafos han sido usados para analizar diversos temas, desde fraudes financieros hasta el rendimiento deportivo, lo que parece indicar que estas bases de datos tienen un alto potencial que en la actualidad no está siendo explotado.

El trabajo descrito en este artículo consolida información de Mercado Libre, una plataforma de comercio electrónico de ventas en línea, con información de vendedores, compradores, productos y las características de estos. Mercado Libre es la compañía argentina de comercio electrónico que se convirtió en la empresa latinoamericana más valiosa en términos de capitalización de mercado (Dinero, 2020). Debido a que no existe actualmente un conjunto de datos de una tienda en línea de productos de tecnología, la presente investigación puede ser de ayuda para una mejor comprensión sobre dichos datos, dado que el funcionamiento de las tiendas en línea es conocido por la gran mayoría de las personas y hace que los datos sean fácilmente entendibles.

Los componentes claves de esta investigación están orientados a entender cómo extraer información de una página web y cómo modelar dicha información en una base de datos de grafos, en este caso, Neo4j, con la cual se puede hacer una representación abstracta que describe la organización de los sistemas de transporte, las interacciones humanas, las telecomunicaciones en redes y las tiendas en línea. A partir de lo anterior, es posible modelar un gran número de estructuras diferentes usando tan solo un paradigma, otorgando así un gran poder a la comunidad de analítica de datos. Para ello, se aplican algoritmos que proporcionan métricas sobre el grafo, los nodos y las relaciones. Estos algoritmos ayudan a detectar los nodos y las comunidades más influyentes en la información.

ESTADO DEL ARTE

Los algoritmos de detección de comunidades se enfocan principalmente en la agrupación de grafos (Scott, 2011) y la influencia de los usuarios (Scott & Carrington, 2011). Rossi y Ahmed (2015) crearon el primer repositorio de datos interactivo con un sitio web para descargar información o aplicar analítica de grafos

sobre conjuntos de datos de diferentes categorías. Adicional a esto, los usuarios pueden discutir cada grafo, así como publicar observaciones y visualizaciones que facilitan la investigación científica.

El trabajo de Vicknair et al. (2010) realiza un comparativo entre una base de datos relacional y una de grafos. En general, la base de datos de grafos funciona mejor en las consultas de tipo estructural que la base de datos relacional. Así mismo, en búsquedas de caracteres de texto completo, la base de datos de grafos es significativamente mejor que la base de datos relacional. Esto se debe a que el mecanismo de indexación utilizado en la base de datos de grafos se basa en cadenas, lo que hace que las consultas numéricas sean menos eficientes. No obstante, es pertinente mencionar que un factor clave en la elección de un sistema de base de datos es la seguridad, frente a lo que la falta de soporte ofrecida por Neo4j se convierte en una limitación (Vicknair et al., 2010).

En las comunidades financieras los fraudes de tipo bancario, seguros y comercio electrónico tienen comportamientos diferentes a los patrones de comportamiento "normal" (Das & Sisk, 2005). Aun así, estos sectores tienen en común que los fraudes se realizan de manera indirecta. Teniendo en cuenta que los estafadores son cada vez más sofisticados al momento de eludir los métodos tradicionales de detección de fraude, trabajando juntos y creando nuevas identidades, los grafos están diseñados para ver las relaciones de los usuarios de forma directa o indirecta, con lo que se ofrece una oportunidad significativa de aumentar los métodos existentes de detección de fraude (Sadowski & Rathle, 2014). Así, se da un nuevo enfoque a los métodos usados para identificar transacciones bancarias irregulares a partir del uso de propiedades de algoritmos de detección de comunidades. Al respecto, de acuerdo con los resultados del estudio de Molloy et al. (2017), es posible reducir sustancialmente los falsos positivos en la puntuación de fraude tradicional. Así mismo, Eboli (2007) presenta un enfoque novedoso para el análisis de impagos de sistemas financieros, representando la información en un grafo con las pérdidas que se propagan en dicha red y calculando las pérdidas e incumplimientos de los usuarios.

En el análisis de las redes sociales el descubrimiento de comunidades es esencial. Por ello, se implementan algoritmos para el descubrimiento de comunidades en Neo4j, evaluando los resultados obtenidos a través de la comparación de métricas de similitud (Kanavos et al., 2017). En este contexto, la descomposición de forma recursiva de un grafo permite el análisis de una red social, evidenciando que el algoritmo de detección de comunidades Louvain crea comunidades mejor distribuidas (Neo4j, 2020a). Por otra parte, se usan algoritmos de centralidad que permiten identificar los nodos más importantes de un grafo. Un ejemplo significativo es identificar la importancia de las páginas web y sus enlaces usando el algoritmo PageRank (Page et al., 1999) para identificar usuarios influyentes en Twitter (Weng et al., 2010) y sus variantes (Kleinberg, 1999).

Otro ejemplo es la popular serie *Game of Thrones*, que cuenta con una gran audiencia, la cual ha sido analizada mediante el esquema de grafos para evidenciar la importancia de los personajes según su participación en cada temporada (Neo4j, 2020b).

De otro lado, la teoría de grafos también se puede implementar en el análisis de rendimiento deportivo con el objetivo de evaluar el rendimiento individual y colectivo de los deportistas. Además, por medio de esta teoría se describen posibles limitaciones de estudios sobre el tema en redes sociales, proporcionando sugerencias para futuras investigaciones en el campo (Ribeiro et al., 2017).

En el área de la salud, la conmoción cerebral relacionada con el deporte es un problema importante de salud pública. Sin embargo, se sabe poco sobre los cambios en redes cerebrales funcionales en adolescentes después de una lesión. Frente a este asunto, Virji-Babul et al. (2014) utilizan la teoría de grafos para evaluar los cambios en las propiedades de la red cerebral después de una conmoción cerebral en atletas adolescentes. Por su parte, Branting et al. (2016) han empleado algoritmos de grafos para identificar riesgo de fraude sanitario, calculando la similitud de actividades de atención médica, procedimientos y prescripción de medicamentos.

CONJUNTO DE DATOS DE MERCADOLIBRE

Los datos del presente estudio fueron extraídos de la página web Mercado Libre Colombia (https://www.mercadolibre.com.co/), actualizada a septiembre de 2020, y corresponden a información de productos, usuarios vendedores, usuarios compradores y opiniones que hacen parte de la categoría "tecnología" en la ciudad de Bogotá.

La información fue obtenida mediante técnicas de *webscraping* con el lenguaje de programación Python, usando la librería Selenium.[1] Con el objetivo de asegurar que la información tuviera un alcance definido, esta fue filtrada a partir del siguiente criterio: información de Bogotá (Colombia) dentro de la categoría "Tecnología, Celulares y Teléfonos, Celulares y Smartphones". De esta manera, se obtuvieron 5.543 registros de productos, 1.096 registros de usuarios vendedores, 15.526 registros de usuarios compradores y 15.997 opiniones.

DESCRIPCIÓN DE LOS DATOS

Inicialmente, se realizó una exploración del conjunto de datos con el objetivo de entender el comportamiento de los productos vendidos y los productos ofertados. En la figura 1 se identifica que las categorías generales más vendidas (a la fecha de consulta) son Xiaomi (9.663 productos) y Samsung (2.170 productos). Comprobando el detalle, en la figura 2 se observa que los productos de categoría Note 8 y Note 8 Pro ocupan los dos primeros lugares.

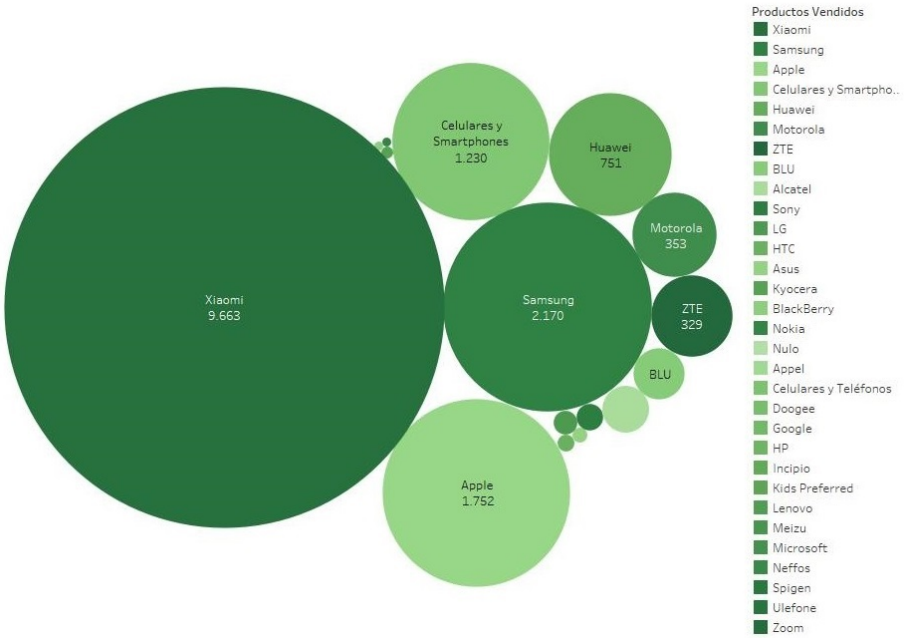


FIGURA 1.
Descripción de los datos: productos vendidos
Fuente: elaboración propia.

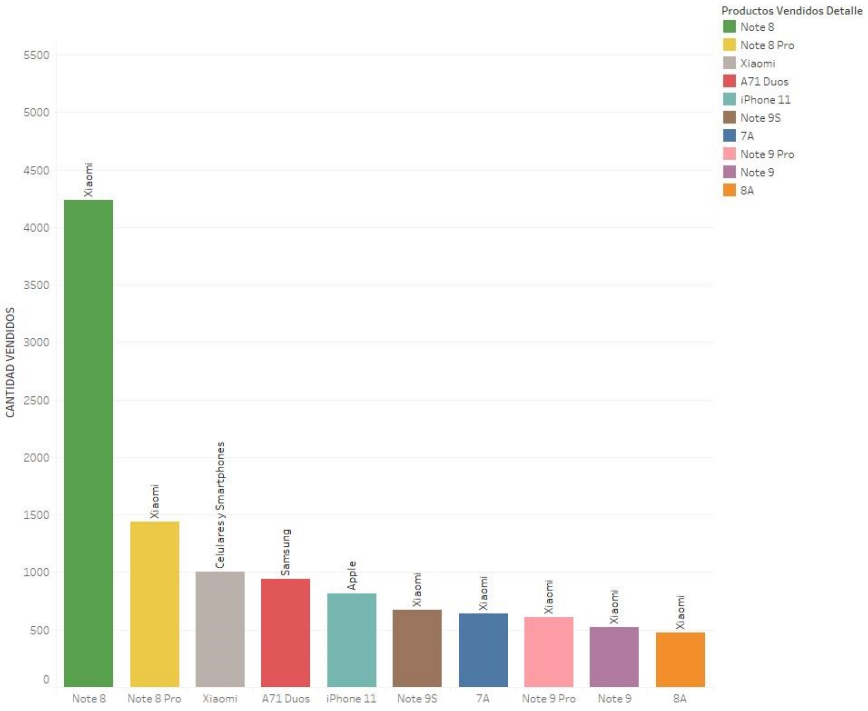


FIGURA 2.
Descripción de los datos: detalle de los productos vendidos
Fuente: elaboración propia.

En la figura 3 se observa que las categorías generales más ofertadas también corresponden a Samsung (1.387 productos) y Xiaomi (1.248 productos). Viendo el detalle de estas categorías, en la figura 4 se aprecia que Xiaomi y Note 8 Pro siguen liderando los dos primeros lugares.

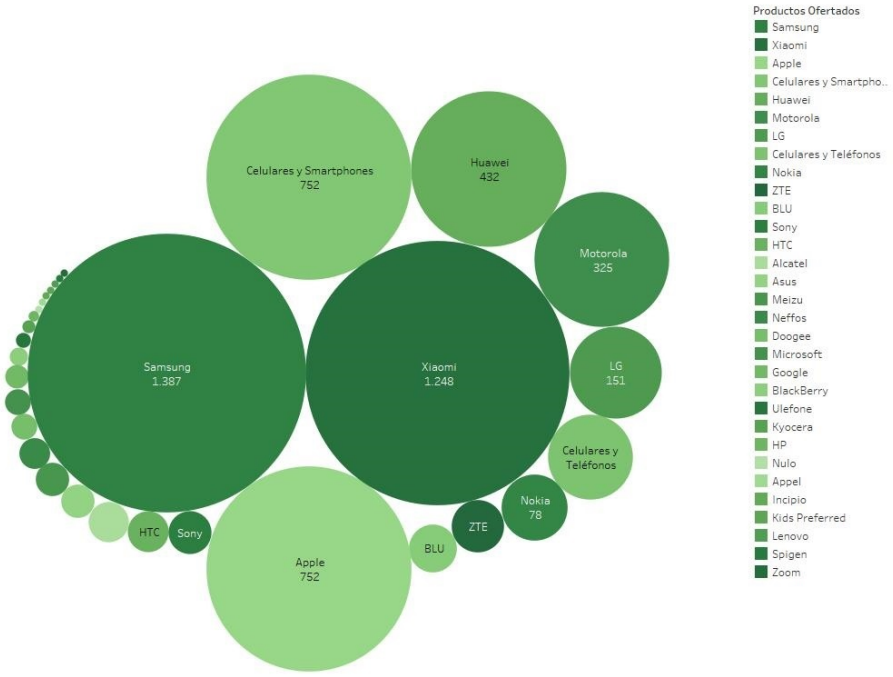


FIGURA 3.
Descripción de los datos: productos ofertados
Fuente: elaboración propia.

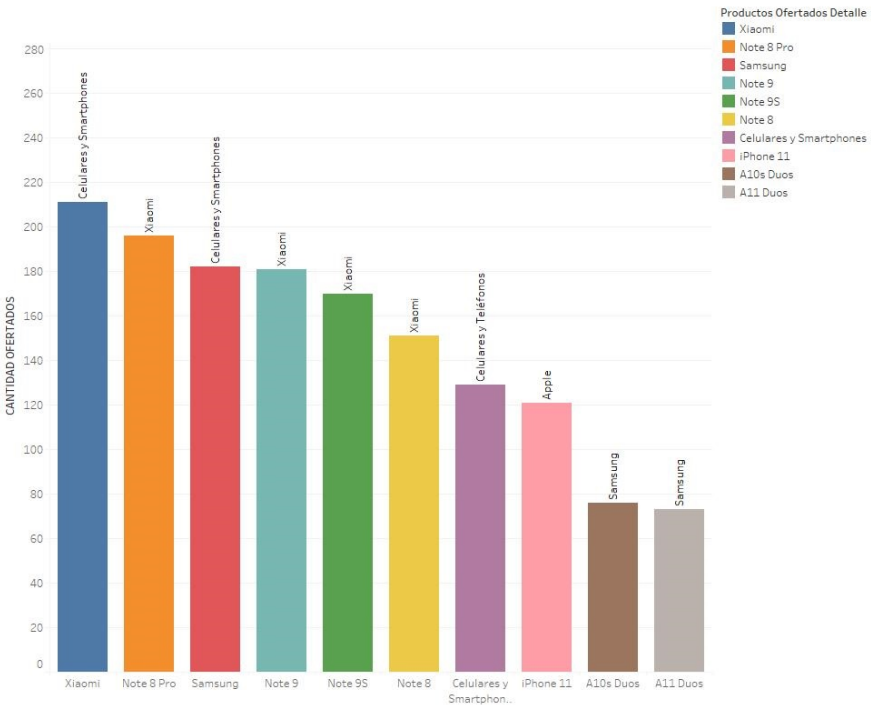
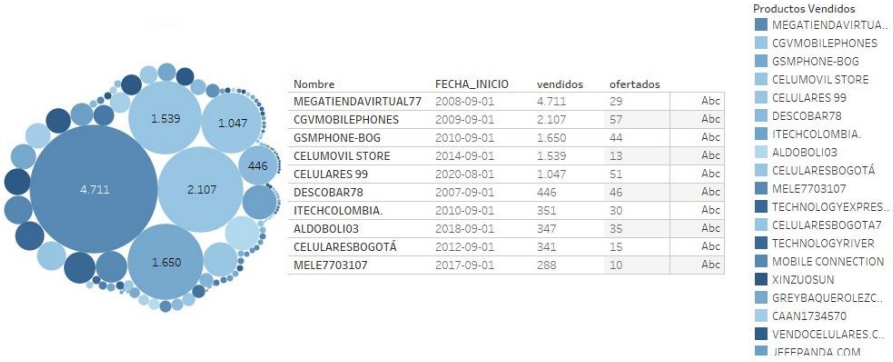


FIGURA 4.
Descripción de los datos: detalle de los productos ofertados
Fuente: elaboración propia.

La figura 5 evidencia que la cantidad de productos vendidos está relacionada con la antigüedad del vendedor en la plataforma de Mercado Libre, como era de esperarse. Sin embargo, una notable excepción es

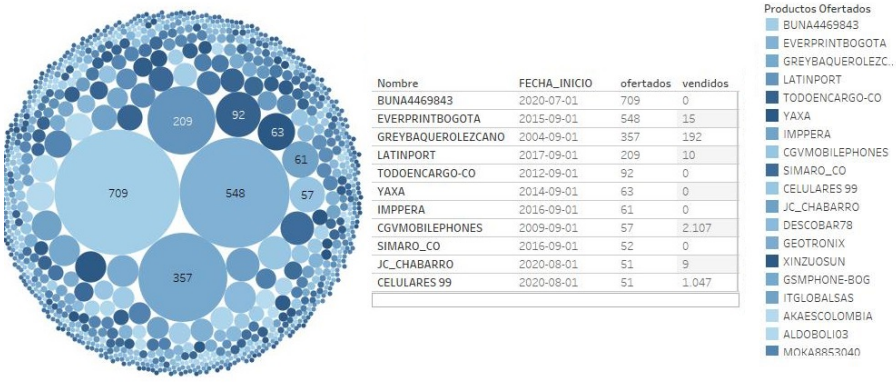
el vendedor “Celulares 99”, quien inició actividades el 1 de agosto de 2020 y se ubica en el puesto número cinco de este ranking.



Nombre	FECHA_INICIO	vendidos	ofertados	
MEGATIENDAVIRTUAL77	2008-09-01	4.711	29	Abc
CGVMOBILEPHONES	2009-09-01	2.107	57	Abc
GSMPHONE-BOG	2010-09-01	1.650	44	Abc
CELUMOVIL STORE	2014-09-01	1.539	13	Abc
CELULARES 99	2020-08-01	1.047	51	Abc
DESCOBAR78	2007-09-01	446	46	Abc
ITECHCOLOMBIA.	2010-09-01	351	30	Abc
ALDOBOLI03	2018-09-01	347	35	Abc
CELULARESBOGOTÁ	2012-09-01	341	15	Abc
MELE7703107	2017-09-01	288	10	Abc

FIGURA 5.
Descripción de los datos: cantidad de productos vendidos
Fuente: elaboración propia.

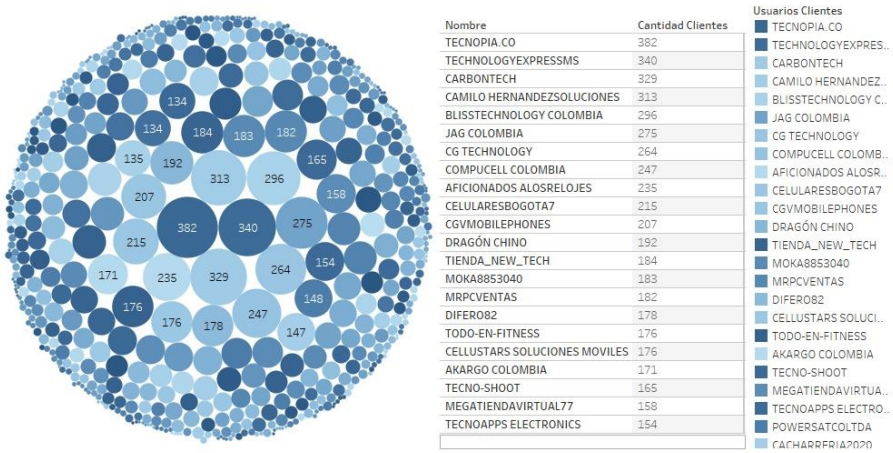
De otro lado, en la figura 6 se observa que la cantidad de productos ofertados no está relacionada con la cantidad de productos vendidos ni con la fecha de inicio del vendedor en Mercado Libre, dado que el vendedor con más productos ofertados inició actividades el 1 de julio de 2020.



Nombre	FECHA_INICIO	ofertados	vendidos
BUNA4469843	2020-07-01	709	0
EVERPRINTBOGOTA	2015-09-01	548	15
GREYBAQUEROLEZCANO	2004-09-01	357	192
LATINPORT	2017-09-01	209	10
TODOENCARGO-CO	2012-09-01	92	0
YAXA	2014-09-01	63	0
IMPPERA	2016-09-01	61	0
CGVMOBILEPHONES	2009-09-01	57	2.107
SIMARO_CO	2016-09-01	52	0
JC_CHABARRO	2020-08-01	51	9
CELULARES 99	2020-08-01	51	1.047

FIGURA 6.
Descripción de los datos: cantidad de productos ofertados
Fuente: elaboración propia.

La figura 7 muestra que existe un conjunto de vendedores constantes en Mercado Libre, lo cual se puede deducir por la cantidad de clientes que estos tienen. En contraste, un mayor grupo de vendedores cuenta con pocos clientes, indicando que se trata de vendedores esporádicos en esta plataforma.



Nombre	Cantidad Clientes
TECNOPIA.CO	382
TECHNOLOGYEXPRESSMS	340
CARBONTECH	329
CAMILO HERNANDEZSOLUCIONES	313
BLISSTECHNOLOGY COLOMBIA	296
JAG COLOMBIA	275
CG TECHNOLOGY	264
COMPUCELL COLOMBIA	247
AFICIONADOS ALOSRELOJES	235
CELULARESBOGOTA7	215
CGVMOBILEPHONES	207
DRAGÓN CHINO	192
TIENDA_NEW_TECH	184
MOKA8853040	183
MRPCVENTAS	182
DIFERO82	178
TODO-EN-FITNESS	176
CELLUSTARS SOLUCIONES MOVILES	176
AKARGO COLOMBIA	171
TECNO-SHOOT	165
MEGATIENDAVIRTUAL77	158
TECNOAPPS ELECTRONICS	154

FIGURA 7.
Descripción de los datos: usuarios vendedores
Fuente: elaboración propia.

ESQUEMA DEL GRAFO

De acuerdo con el modelo de datos de Neo4j, un grafo es una composición de un conjunto de objetos, conocidos como nodos, que se relacionan con otros (nodos) a través de un conjunto de conexiones, conocidas como aristas (Neo4j, 2020c). Los nodos se utilizan para representar entidades, mientras que las etiquetas sirven para agrupar nodos en conjuntos donde todos los nodos que tienen determinada etiqueta pertenecen al mismo conjunto. Así, un nodo puede estar asociado a cero, una o varias etiquetas. De otro lado, una relación expresa conexión entre dos nodos. En Neo4j, cada relación tiene exactamente un tipo. Las relaciones, a su vez, organizan a los nodos en estructuras, lo que permite que un grafo represente una lista, un árbol, un mapa o una entidad compuesta (Neo4j, 2020c).

En la figura 8 se definen nodos y relaciones que comprenden el modelo de datos del conjunto de datos de Mercado Libre, mientras que la tabla 1 presenta el respectivo diccionario de datos. Los nodos con etiqueta DEPARTAMENTO representan los departamentos registrados en el sistema. Los nodos con etiqueta MUNICIPIO se conectan con el departamento mediante la relación MUNICIPIO_EN. Los nodos con etiqueta USUARIO contiene los datos de vendedores y compradores, donde un usuario puede tener uno o ambos roles. Los usuarios están situados en un municipio (a través de la relación UBICADO_EN), opinan sobre otros usuarios (a través de la relación OPINA) y venden productos (a través de la relación VENDE). Los nodos con etiqueta PRODUCTO contienen los datos de los productos ofertados por los vendedores y están clasificados dentro de una categoría mediante la relación PERTENECE. Por último, los nodos con etiqueta CATEGORIA clasifican a los productos, considerando que una categoría puede estar clasificada dentro de otra, generando una relación jerárquica (relación CLASIFICADO_EN).

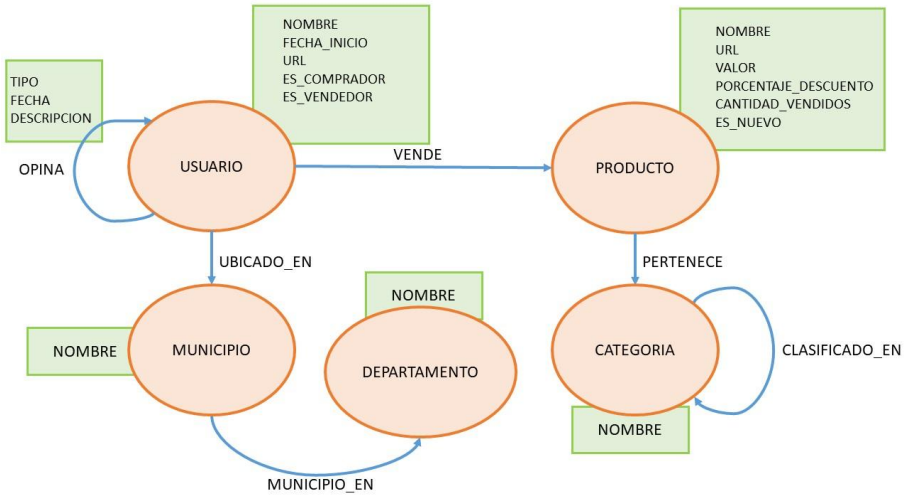


FIGURA 8.
Esquema de grafos
Fuente: elaboración propia.

TABLA 1.
Diccionario de datos

Tipo objeto	Nombre objeto	Descripción objeto	Nombre campo	Descripción campo	Tipo dato
Nodo	DEPARTAMENTO	Departamento donde los usuarios están ubicados	CODIGO	Código unico del departamento.	STRING
			NOMBRE	Nombre del departamento.	STRING
Nodo	MUNICIPIO	Municipio donde los usuarios están ubicados	CODIGO	Código unico del municipio.	STRING
			NOMBRE	Nombre del municipio.	STRING
Nodo	USUARIO	Usuarios vendedores o compradores	NOMBRE	Nombre del usuario.	STRING
			FECHA_INICIO	Fecha de inicio del usuario en el sistema.	DATE
			URL	URL del perfil del usuario.	STRING
			ES_COMPRADOR	Identifica si el usuario es comprador.	BOOLEAN
			ES_VENDEDOR	Identifica si el usuario es vendedor.	BOOLEAN
Nodo	PRODUCTO	Productos ofertados.	NOMBRE	Nombre del producto.	STRING
			URL	URL del producto ofertado.	STRING
			VALOR	Valor del producto.	STRING
			PORCENTAJE_DES	Porcentaje de descuanto del producto.	STRING
			CANTIDAD_VENDI	Cantidad de productos vendidos.	STRING
			ES_NUEVO	Identifica si el producto es nuevo.	BOOLEAN
Nodo	CATEGORIA	Clasificación de los productos	CODIGO	Código unico del departamento.	STRING
			NOMBRE	Nombre del departamento.	STRING
Relación	MUNICIPIO_EN	Relación entre los nodos MUNICIPIO y DEPARTAMENTO			
Relación	UBICADO_EN	Relación del municipio donde se ubica un usuario.			
Relación	OPINA	Opinión que da un usuario comprador a un usuario vendedor.	TIPO	Tipo de opinión.	STRING
			FECHA	Fecha de opinión.	DATE
			DESCRIPCION	Descripción de opinión.	STRING
Relación	VENDE	Relación entre el producto ofertado y el usuario vendedor.			
Relación	PERTENECE	Relación que clasifica a un producto.			
Relación	CLASIFICADO_EN	Relación jerarquica de las categorias.			

Fuente: elaboración propia.

IMPORTACIÓN DE LOS DATOS EN NEO4J

La información recolectada con técnicas de webscraping fue guardada en forma tabular usando archivos con formato CSV. Neo4j, con su lenguaje Cypher, proporciona una opción para facilitar la importación de los datos contenidos en los archivos a una estructura de grafos.

Es importante desarrollar los siguientes pasos para cargar los datos a Neo4j:

§ Contar con un motor de bases de datos Neo4j instalado y en correcto funcionamiento.

§ Los datos se pueden descargar en la URL https://github.com/gersongelvez/TESIS_MAESTRIA/tree/master/DATOS (Carrillo-Gelvez, 2019).

§ Usar la herramienta de importación de archivos con formato CSV.

§ Ejecutar los comandos de importación uno a uno en la consola de Neo4j.

APLICACIÓN DE CONSULTAS EN LENGUAJE CYPHER

Neo4j permite la realización de consultas usando Cypher, un lenguaje declarativo inspirado en SPARQL (Pérez et al., 2009) que permite expresar la coincidencia de patrones en grafos. En esta sección se presentan algunas consultas sobre el esquema del grafo definido para Mercado Libre con el fin de ilustrar las funcionalidades de Cypher.

Listado de departamentos y municipios

La consulta en lenguaje Cypher para listar los departamentos y municipios es:

MATCH (M:MUNICIPIO)-[:MUNICIPIO_EN]->(D:DEPARTAMENTO) RETURN D.NOMBRE AS DEPARTAMENTO, M.NOMBRE AS MUNICIPIO

Esta consulta busca el patrón en el grafo que consisten en un nodo M con la etiqueta MUNICIPIO y un nodo D con la etiqueta DEPARTAMENTO, que se conectan a través de una relación de tipo MUNICIPIO_EN.

La tabla 2 muestra que existen departamentos y municipios diferentes a Bogotá, lo que quiere decir que a pesar de que solo se recolectó información de productos de tecnología en Bogotá hay vendedores de diferentes partes de Colombia dentro de esta categoría. Además, la página de Mercado Libre registra a Bogotá como un departamento y a sus localidades como municipios, por lo que resulta necesario aplicar un proceso de calidad de datos para estandarizar la información.

TABLA 2.
Departamentos y municipios

Departamento	Municipio
ATLÁNTICO	BARRANQUILLA
BOGOTÁ DC	USME
BOGOTÁ DC	USAQUÉN
BOGOTÁ DC	TUNJUELITO
BOGOTÁ DC	TEUSAQUILLO
BOGOTÁ DC	SUMAPAZ
BOGOTÁ DC	SUBA
BOGOTÁ DC	SANTA FE
BOGOTÁ DC	SAN CRISTÓBAL SUR
BOGOTÁ DC	RAFAEL URIBE URIBE
BOGOTÁ DC	PUENTE ARANDA
BOGOTÁ DC	MÁRTIRES
BOGOTÁ DC	LA CANDELARIA
BOGOTÁ DC	KENNEDY
BOGOTÁ DC	FONTIBÓN
BOGOTÁ DC	ENGATIVÁ
BOGOTÁ DC	CIUDAD BOLÍVAR
BOGOTÁ DC	CHAPINERO
BOGOTÁ DC	BOSA
BOGOTÁ DC	BARRIOS UNIDOS
BOGOTÁ DC	ANTONIO NARIÑO
CESAR	AGUACHICA
CUNDINAMARCA	SOACHA
CUNDINAMARCA	MOSQUERA
CUNDINAMARCA	FUNZA

Fuente: elaboración propia.

Cantidad de usuarios por departamento y municipio

La consulta en lenguaje Cypher para listar los 10 municipios y su respectivo departamento con más usuarios vendedores es:

MATCH (U:USUARIO)-[:UBICADO_EN]->(M:MUNICIPIO)-[:MUNICIPIO_EN]->(D:DEPARTAMENTO) RETURN D.NOMBRE AS DEPARTAMENTO, M.NOMBRE AS MUNICIPIO, COUNT(U) AS CANTIDAD ORDER BY CANTIDAD DESC LIMIT 10

Esta consulta muestra el uso de la función de agregación COUNT, similar a la del lenguaje SQL, que en este caso realiza un conteo de los usuarios U y les asigna el alias CANTIDAD. El resultado se ordena de forma descendente y se limita a diez filas.

La tabla 3 muestra que el municipio de Suba contiene la mayor cantidad de usuarios vendedores. Por otro lado, se identifica que los nombres de los municipios necesitan un proceso de calidad de datos debido a algunos errores de ortografía en sus nombres.

TABLA 3.
Usuarios por departamento y municipio

Departamento	Municipio	Cantidad
BOGOTÁ DC	SUBA	130
BOGOTÁ DC	KENNEDY	115
BOGOTÁ DC	ENGATIVA	96
BOGOTÁ DC	CHAPINERO	86
BOGOTÁ DC	USAQUÉN	85
BOGOTÁ DC	PUENTE ARANDA	58
BOGOTÁ DC	BOSA	48
BOGOTÁ DC	FONTIBÓN	47
BOGOTÁ DC	TEUSAQUILLO	46

Fuente: elaboración propia.

Productos de Apple más vendidos

La consulta en lenguaje Cypher para listar los 5 productos de Apple más vendidos y sus características es:

MATCH(U:USUARIO)-[:VENDE]->(P:PRODUCTO)-[:PERTENECE]->(C:CATEGORIA) WHERE C.CODIGO CONTAINS 'Apple' RETURN U.NOMBRE AS VENDEDOR, P.NOMBRE AS PRODUCTO, C.NOMBRE AS CATEGORIA, toInteger(P.VALOR)/3703.7 AS "VALOR (USD)", toInteger(P.CANTIDAD_VENDIDOS) AS CANTIDAD_VENDIDOS ORDER BY CANTIDAD_VENDIDOS DESC LIMIT 5

En esta consulta la cláusula WHERE permite filtrar por atributos de los nodos, en este caso estipulado que solo se deben incluir usuarios que venden productos de la categoría Apple. La tabla 4 indica que el vendedor "Celulares 99" es quien ha vendido más productos de la categoría Apple.

TABLA 4.
Productos de Apple más vendidos

Vendedor	Producto	Categoría	Valor (USD)	Cantidad vendidos
Celulares 99	Celular iPhone 11 128gb nuevo 100% original y sellada	iPhone 11	943,81	203
Celulares 99	Celular iPhone 11 64gb sellado nuevo 4g	iPhone 11	850,77	193
Mobile Connection	iPhone 7 32gb 4g Lte 12mp 4k 3d touch Garantía + obsequios	iPhone 7	306,46	114
Celulares 99	Celular iPhone SE 2020 64gb chip A13	SE	533,58	100
Gmsphone-bog	iPhone 11 64gb sellado entrega inmediata 1 año garantia	iPhone 11	848,28	80

Fuente: elaboración propia.

Categorías con el máximo número de niveles

La consulta en lenguaje Cypher para mostrar las cinco categorías con el máximo número de niveles es:

MATCH path = (HIJA:CATEGORIA)-[:CLASIFICADO_EN *1..]->(PADRE:CATEGORIA) RETURN PADRE.NOMBRE AS CATEGORIA, MAX(length(path)) as depth ORDER BY depth DESC, CATEGORIA LIMIT 5

Esta consulta busca el patrón en el grafo donde aparece un camino o una (o más) arista del tipo CLASIFICADO_EN y lo asigna a la variable *path*. Posteriormente, la función length permite contar la longitud del camino en el grafo. Así, la tabla 5 indica que el máximo número de niveles que tiene la jerarquía de categorías es cuatro.

TABLA 5.
Categorías con el máximo número de niveles

Categoría	Niveles
Celulares y Teléfonos	4
Celulares y Smartphones	3
Alcatel	2
Apple	2
Apple	2

Fuente: elaboración propia.

APLICACIÓN DE ANALÍTICA DE GRAFOS

La librería Graph Data Science (GDS)[2] de Neo4j ofrece una serie de algoritmos para aplicar analítica de grafos sobre el conjunto de datos de Mercado Libre. En esta sección se describen los conceptos que se deben tener en cuenta para ello.

Algoritmos

Los algoritmos se utilizan para calcular métricas de grafos, nodos o relaciones. Estos pueden proporcionar información sobre entidades relevantes en el grafo (centralidad y clasificación) o estructuras inherentes, como la detección de comunidades, particiones de grafos y agrupación. Con frecuencia, muchos algoritmos atraviesan el grafo de manera iterativa utilizando recorridos aleatorios, búsquedas de amplitud, búsquedas de profundidad o coincidencia de patrones. Debido al crecimiento exponencial de las posibles rutas al aumentar la distancia, muchos de los enfoques también tienen una alta complejidad algorítmica. Afortunadamente, existen algoritmos optimizados que utilizan ciertas estructuras del grafo, memorizan partes ya exploradas y paralelizan operaciones (Neo4j, 2020a).

Catálogo del grafo

Para ejecutar los algoritmos de la manera más eficiente posible, la biblioteca GDS utiliza la memoria RAM para representar los datos del grafo. Por lo tanto, es necesario cargar los datos en la memoria. La cantidad de datos cargados se puede controlar mediante las llamadas proyecciones de grafos, que también permiten, por ejemplo, filtrar por etiquetas de nodos y tipos de relaciones, entre otras opciones.

Modo de ejecución de algoritmos

Se deben tener en cuenta los modos de ejecución Stream, Stats, Mutate o Write, los cuales se describen a continuación:

§ Stream: devuelve los resultados en memoria de la consulta ejecutada. Esto es similar a como funcionan las consultas estándares de Neo4j con su lenguaje Cypher.

§ Stats: devuelve resultados estadísticos de la ejecución del algoritmo, como recuentos o distribuciones de percentiles. Este modo no realiza modificaciones o actualizaciones sobre los datos.

§ Mutate: vuelve a escribir los resultados del cálculo del algoritmo en el grafo que está en memoria.

§ Write: vuelve a escribir los resultados del cálculo del algoritmo en la base de datos de Neo4j. Este modo de ejecución fue el usado en los algoritmos aplicados a los datos de Mercado Libre.

Sintaxis

La sintaxis empleada para la ejecución de algoritmos de la librería GDS es la siguiente:

```
CALL gds.<algoritmo>.write({
nodeProjection: <nombre del nodo>,
relationshipProjection: {
<relación>: {
type: <relación>,
orientation: <orientación> +
}
},
writeProperty: <nombre de la propiedad>
})
```

En la sintaxis anterior se deben asignar los valores respectivos al momento de ejecutar un algoritmo. El significado de los valores que se deben asignar en cada ejecución es:

§ <algoritmo>: nombre del algoritmo de la librería GDS.

§ <nombre del nodo>: nombre del nodo sobre el que se va a aplicar el algoritmo.

§ <relación>: relación que conecta a los nodos.

§ <orientación>: orientación de la relación. Los valores permitidos para esta propiedad son NATURAL (se toma la orientación definida en el grafo), UNDIRECTED (no se tiene en cuenta la orientación de la relación) y REVERSE (se toma la dirección inversa definida en el grafo).

§ <nombre de la propiedad>: campo donde queda guardado el resultado del algoritmo.

Los tipos de algoritmos que se pueden ejecutar en la librería GDS se describen a continuación.

Algoritmos de centralidad

Los algoritmos de centralidad se utilizan para descubrir los roles de nodos en un grafo y su impacto en la red. Estos identifican nodos importantes y ayudan a comprender la credibilidad, accesibilidad y velocidad a la que se propagan las cosas, así como los puentes entre los grupos. Estos algoritmos fueron creados para analizar redes sociales, pero han sido de utilidad en diferentes casos de uso. La figura 9 presenta algunos tipos de algoritmos de centralidad.

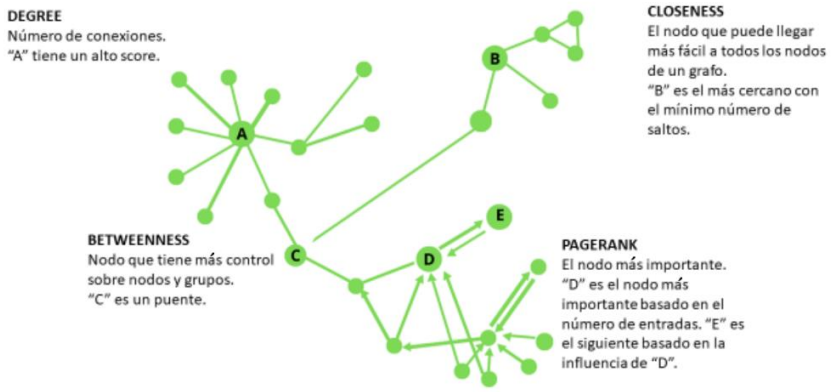


FIGURA 9.
Algoritmos de centralidad
Fuente: elaboración propia.

Algoritmo de centralidad *Degree*

Se puede aplicar el algoritmo de centralidad *Degree* sobre las categorías de los productos para identificar cuál categoría tiene más conexiones u oferta mayor número de productos. El código fuente en Cypher aplicado es:

```
CALL gds.degree.write({
nodeProjection: 'CATEGORIA',
relationshipProjection: {
CLASIFICADO_EN: {
type: 'CLASIFICADO_EN',
orientation: 'UNDIRECTED'
}
},
writeProperty: 'DEGREE_SCORE'
})
```

La tabla 6 permite identificar que la categoría Samsung es la más importante según el número de conexiones que tiene con los demás nodos.

TABLA 6.
Resultado del algoritmo de centralidad Degree sobre las categorías

Categoría	DEGREE_SCORE
Samsung	367
Xiaomi	287
Celulares y Teléfonos | Celulares y Smartphones	235
Xiaomi | Note 8 Pro	197
Xiaomi | Note 9	182
Xiaomi | Note 9S	171
Xiaomi | Note 8	152
Apple | iPhone 11	122
Huawei	113
Motorola	110

Fuente: elaboración propia.

Algoritmo de centralidad *Weighted Degree*

Este algoritmo nos ayuda a identificar el nodo más importante del grafo con base en el número de conexiones existentes, asignando un peso a cada conexión. Se aplica el algoritmo a las categorías, donde el peso de cada conexión es el número de productos vendidos. El código fuente en Cypher aplicado es:

```
CALL gds.degree.write({
nodeProjection: 'CATEGORIA',
relationshipProjection: {
CLASIFICADO_EN: {
type: 'CLASIFICADO_EN',
orientation: 'UNDIRECTED',
properties: 'CANTIDAD'
}
},
relationshipWeightProperty: 'CANTIDAD',
writeProperty: 'WEIGHT_DEGREE_SCORE'
})
```

En la tabla 7 se puede identificar que la categoría Xiaomi|Note8 es la más importante, según el número de conexiones con los demás nodos y la cantidad de productos vendidos.

TABLA 7.
Resultado del algoritmo de centralidad *Weighted Degree* sobre las categorías

Código	WEIGHT_DEGREE_SCORE
Xiaomi | Note 8	4247
Xiaomi | Note 8 Pro	1444
Xiaomi	1099
Samsung | A71 Duos	941
Apple | iPhone 11	819
Xiaomi | Note 9S	686
Xiaomi | 7A	642
Xiaomi | Note 9 Pro	611
Xiaomi | Note 9	530
Xiaomi | 8A	476

Fuente: elaboración propia.

Algoritmo de centralidad *PageRank*

Con este algoritmo se puede identificar la categoría más importante de acuerdo con el número de conexiones, tomando en cuenta la importancia de las categorías que la referencian. El código fuente en Cypher aplicado es:

```
CALL gds.pageRank.write({
nodeProjection: "CATEGORIA",
relationshipProjection: "CLASIFICADO_EN",
maxIterations: 20,
dampingFactor: 0.85,
writeProperty: 'PAGE_RANK_SCORE'
})
```

Según la información en la tabla 8, las categorías "Celulares y Smartphones" y "Celulares y Teléfonos" tiene los dos puntajes más altos, debido a que son los nodos padres de todas las categorías y toman la importancia de las categorías hijas.

TABLA 8.
Resultado del algoritmo de centralidad *PageRank* sobre las categorías

Código	PAGE_RANK_SCORE
Celulares y Smartphones	616
Celulares y Teléfonos	524
Samsung	197
Xiaomi	172
Apple	86

Fuente: elaboración propia.

Algoritmos de detección de comunidades

La formación de comunidades es frecuente en los esquemas de grafos debido a que esta ayuda a identificar el comportamiento grupal. Los miembros de cada comunidad tienen más relaciones dentro del grupo que con nodos fuera de él. Esto revela grupos de nodos, grupos aislados y la estructura de la red. Los algoritmos de detección de comunidades son usados para visualizar la red con fines de inspección general, como los expuestos en la figura 10.

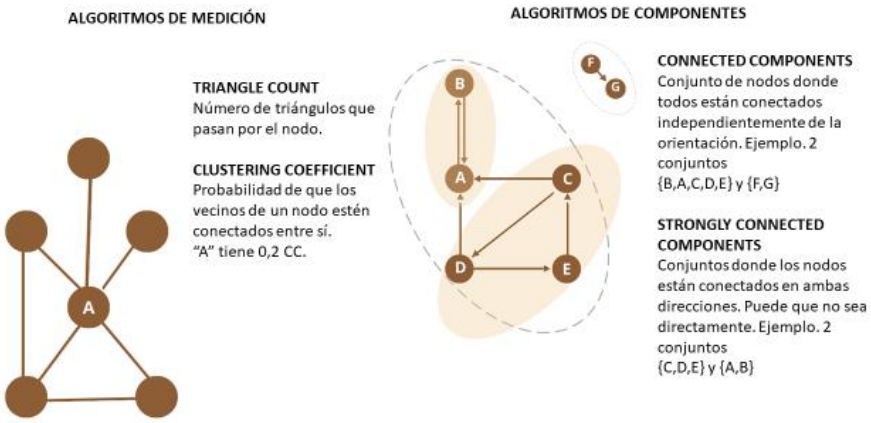
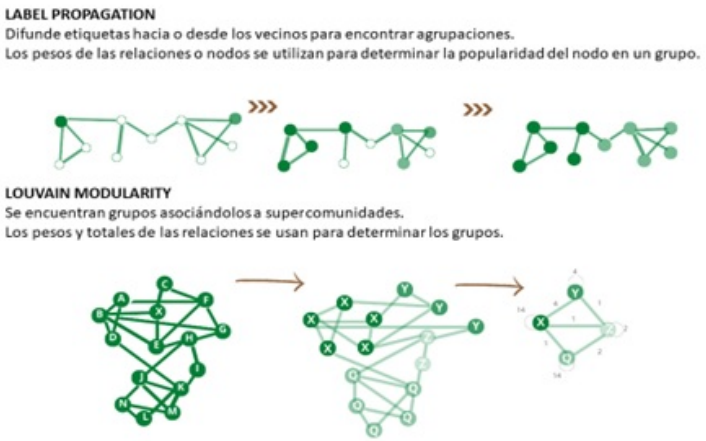


FIGURA 10.
Algoritmos de detección de comunidades



Fuente: elaboración propia.

Algoritmo de detección de comunidades *Connected Components*

Se aplica este algoritmo a los usuarios vendedores y compradores con el fin de identificar grupos de usuarios conectados entre sí. El código fuente en Cypher aplicado es:

```
CALL gds.wcc.write({
nodeProjection: 'USUARIO',
relationshipProjection: {
OPINA: {
type: 'OPINA',
orientation: 'UNDIRECTED'
```

```
}
},
writeProperty: 'WCC_COMUNIDAD'
})
```

Se identifica en la tabla 9 que existe una comunidad con 13.594 usuarios y 4 comunidades con muy pocos usuarios. Esto indica que la mayoría de los usuarios tiene el mismo comportamiento y las mismas conexiones.

TABLA 9.
Resultado del algoritmo de detección de comunidades *Connected Components*

Comunidad	N.° usuarios
1	13.594
2	52
3	51
4	47
5	45

Fuente: elaboración propia.

Algoritmo de detección de comunidades *Label Propagation*

Se aplica este algoritmo a los usuarios vendedores y compradores para identificar grupos de usuarios que estén conectados entre sí usando las conexiones de los vecinos. El código fuente en Cypher aplicado es:

```
CALL gds.labelPropagation.write({
nodeProjection: 'USUARIO',
relationshipProjection: {
OPINA: {
type: 'OPINA',
orientation: 'UNDIRECTED'
}
},
writeProperty: 'LP_COMUNIDAD'
})
```

En la tabla 10 se identifica que este algoritmo distribuye de manera uniforme las comunidades de usuarios.

TABLA 10.
Resultado del algoritmo de detección de comunidades *Label Propagation*

Comunidad	N.° usuarios
1	382
2	339
3	327
4	326
5	306

Fuente: elaboración propia.

Se aplica este algoritmo a los usuarios vendedores y compradores para identificar supercomunidades. El código fuente en Cypher aplicado es:

```
CALL gds.louvain.write({
nodeProjection: 'USUARIO',
relationshipProjection: {
OPINA: {
type: 'OPINA',
orientation: 'UNDIRECTED'
}
},
writeProperty: 'LOUV_COMUNIDAD'
})
```

En la tabla 11 se puede observar que este algoritmo distribuye de manera más uniforme las comunidades de usuarios, en comparación con los algoritmos anteriores.

TABLA 11.
Resultado del algoritmo de detección de comunidades *Louvain Modularity*

Comunidad	N.° usuarios
1	393
2	341
3	339
4	337
5	328

Fuente: elaboración propia.

Visualización de grafos

Usando los resultados de los algoritmos aplicados en las secciones anteriores se puede visualizar la importancia de los nodos y las comunidades identificadas. Para poder visualizar las estadísticas que proporcionan los algoritmos de Neo4j puede utilizarse la librería Neovis.js.[3]

Visualización de categorías de Mercado Libre

Usando las métricas obtenidas en los puntos anteriores, la figura 11 permite visualizar las comunidades identificadas para las categorías de productos de Mercado Libre, donde el tamaño del nodo indica la importancia dentro del grafo.

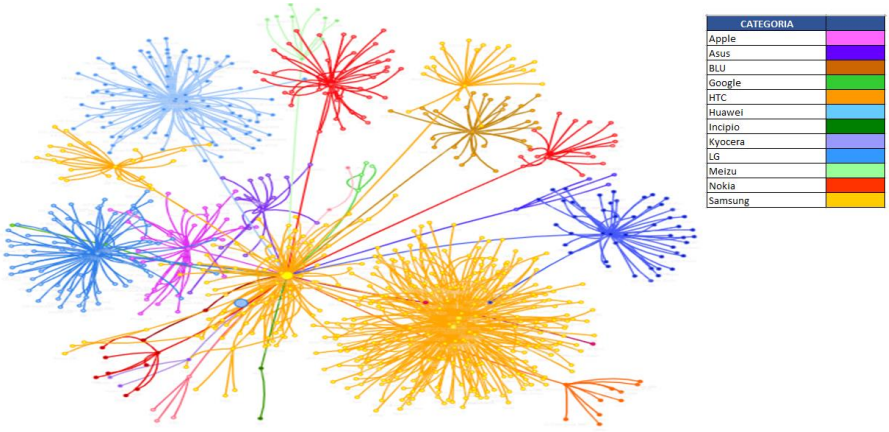


FIGURA 11.
Visualización de comunidades de las categorías de Mercado Libre
Fuente: elaboración propia.

Visualización de usuarios de Mercado Libre

Usando las métricas obtenidas en los puntos anteriores, la figura 12 indica que existen comunidades con una gran cantidad de usuarios pero pocas conexiones con otras comunidades. En contraste, hay comunidades más pequeñas que sí se relacionan con otras comunidades.

FIGURA 12.
Visualización de comunidades de los usuarios de MercadoLibre
Fuente: elaboración propia.

CONCLUSIONES

Se recolectaron datos públicos de Mercado Libre para productos ofertados en la ciudad de Bogotá, aunque los usuarios vendedores no necesariamente están en la misma ciudad donde se ofertan tales productos, puesto que se identificó la presencia de vendedores provenientes de los departamentos de Atlántico, César y Cundinamarca. Además, Bogotá está registrado en el sistema como un departamento, mientras que sus localidades figuran como municipios, siendo Suba aquel con más usuarios vendedores.

Gracias a la notación intuitiva de nodos y relaciones, el lenguaje Cypher es mucho más sencillo que SQL; más aún si se aumenta el número de *joins* o un patrón de búsqueda. Además, el lenguaje SQL resulta ser menos eficiente, al tener que recorrer todas las tablas mediante *joins*.

Con el método empleado se pueden detectar los casos de baja oferta y alta demanda para un producto, como sucede con el dispositivo Xiaomi Note 8, que resultó ser el producto más vendido (figura 2), a pesar de que ocupa el sexto lugar dentro de los productos más ofertados (figura 4).

El usuario vendedor "Celulares 99" ingresó a la plataforma el 1 de agosto de 2020 y ha vendido 1.047 productos. Por su parte, el usuario vendedor "Descobar78" ingresó el 1 de septiembre de 2007 y ha logrado

vender 446 productos. Haciendo esta comparación, "Celulares 99" es un usuario con un buen perfil de ventas para Mercado Libre, debido a que en pocos meses ha logrado alcanzar ventas que otros usuarios no han alcanzado en años de operación.

Las consultas en lenguaje declarativo Cypher revelan información de cantidades de productos, vendedores, compradores y categorías, mientras que los algoritmos de analítica de grafos son adecuados para revelar patrones de la relación entre vendedores y compradores sin estar directamente conectados. Por ello, la visualización de grafos contribuye a dar sentido al volumen de usuarios y las categorías de productos, además de identificar comunidades que no son visibles a simple vista.

La información esquematizada debe tener características puntuales para que un algoritmo de grafos permita obtener un buen resultado. Esto se pudo observar al aplicar el algoritmo de detección de comunidades *Connected Components*, puesto que no se distribuyó la información en comunidades homogéneas, dando como resultado que la mayoría de los usuarios fueran clasificados dentro de una misma comunidad.

En conclusión, la analítica de grafos aplicada a los datos de Mercado Libre ayuda a identificar las categorías ofertadas más importantes, así como los usuarios más influyentes dentro del sistema. De igual forma, la aplicación de este método hace que se pueda categorizar la información en comunidades, tomando como criterio características similares que no se identifican a simple vista.

Referencias

Branting, L. K., Reeder, F., Gold, J., & Champney, T. (2016). Graph analytics for healthcare fraud risk estimation. 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 845-851). ASONAM. https://www.computer.org/csdl/proceedings-article/asonam/2016/07752336/12OmNz4SOvm

Carrillo-Gelvez, G. (2019). Conjunto de datos de Mercadolibre [data set]. GitHub. https://github.com/gersongelvez/TESIS_MAESTRIA/tree/master/DATOS

Das, S. R., & Sisk, J. (2005). Financial communities. Journal of Portfolio Management, 31(4), 112-123.

DB-Engines (2020a). DB-Engines ranking. https://db-engines.com/en/ranking

DB-Engines (2020b). DB-Engines ranking of graph DBMS. https://db-engines.com/en/ranking/graph+dbms

Dinero. (2020, agosto 10). Mercado Libre: ¿cómo llegó a ser la firma más valiosa de Latinoamérica? https://www.dinero.com/empresas/articulo/mercado-libre-es-la-empresa-mas-valiosa-de-america-latina-en-2020/295269

Eboli, M. (2007). Systemic risk in financial networks: A graph-theoretic approach. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.466.7515&rep=rep1&type=pdf

Kanavos, A., Drakopoulos, G., & Tsakalidis, A. (2017). Graph community discovery algorithms in Neo4j with a regularization-based evaluation metric. Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017) (pp. 403-410). WEBIST. https://www.scitepress.org/papers/2017/63821/63821.pdf

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5), 604-632. https://doi.org/10.1145/324133.324140

Molloy, I., Chari, S., Finkler, U., Wiggerman, M., Jonker, C., Habeck, T., Park, Y., Jordens, F., & van-Schaik, R. (2017). Graph analytics for real-time scoring of cross-channel transactional fraud. En J. Grossklags & B. Preneel (eds.), Financial Cryptography and Data Security (pp. 22-40). Springer. https://doi.org/10.1007/978-3-662-54970-4_2

Neo4j. (2020a). Neo4j Graph Data Science Library. https://neo4j.com/graph-data-science-library/

Neo4j. (2020b). The Graph of Thrones [Season 7 Contest]. https://neo4j.com/blog/graph-of-thrones/

Neo4j (2020c). Neo4j Graph database concepts. https://neo4j.com/docs/getting-started/current/graphdb-concepts/#graphdb-concepts

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank Citation Ranking: Bringing order to the web. Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/

Pragma (2018). Los beneficios de las bases de datos NoSQL. https://www.pragma.com.co/blog/los-beneficios-de-las-bases-de-datos-nosql

Ribeiro, J., Silva, P., Duarte, R., Davids, K., & Garganta, J. (2017). Team sports performance analysed through the lens of social network theory: Implications for research and practice. Sports Medicine, 47(9), 1689-1696.

Rossi, R. A., & Ahmed, N. K. (2015). The network data repository with interactive graph analytics and visualization. arXiv, 2. http://arxiv.org/abs/1410.3560

Sadowski, G., & Rathle, P. (2014). Fraud detection: Discovering connections with graph databases. Neo4j. https://neo4j.com/whitepapers/white-paper-fraud-detection/

Scott, J., & Carrington, P. (2011). The SAGE handbook of social network analysis. SAGE.

Scott, J. (2011). Social network analysis: Developments, advances, and prospects. SOCNET, 1, 21-26. https://doi.org/10.1007/s13278-010-0012-6

Svensson, J. (2020). SDTimes. https://sdtimes.com/databases/guest-view-relational-vs-graph-databases-use/

Pérez, J., Arenas, M., & Gutierrez, C. (2009). Semantics and complexity of SPARQL. ACM Transactions on Database Systems, 34(3), 1-45.

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., & Wilkins, D. (2010). A comparison of a graph database and a relational database. Proceedings of the 48th Annual Southeast Regional Conference (pp. 1-6). ACM. https://doi.org/10.1145/1900008.1900067

Virji-Babul, N., Hilderman, C. G. E., Makan, N., Liu, A., Smith-Forrester, J., Franks, C., & Wang, Z. J. (2014). Changes in functional brain networks following sports-related concussion in adolescents. Journal of Neurotrauma, 31(23), 1914-1919.

Weng, J., Lim, P., Jiang, J., & He, Q. (2010). Twitterrank: Finding topic-sensitive influential Twitterers. Proceedings of the third ACM International Conference on Web Search and Data Mining (261-270). ACM.

Notas

1 https://pypi.org/project/selenium/
2 https://neo4j.com/graph-data-science-library/
3 https://github.com/neo4j-contrib/neovis.js/