Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

MODELADO BASADO EN GRAFOS PARA ANALIZAR Y SIMULAR REDES DE CORRUPCIÓN

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: Ing. Oscar René Medina Arriola

Director: Dr. José Francisco Cervantes Alvarez

Codirector: Mtro. Víctor Hugo Ortega Guzmán

Tlaquepaque, Jalisco. 13 de julio de 2023

AGRADECIMIENTOS

El autor desea expresar su más profundo agradecimiento a sus asesores de TOG, el Maestro Víctor Ortega y el Dr. Francisco Cervantes, cuya invaluable orientación y apoyo han sido fundamentales para el desarrollo y culminación de este proyecto. Además, se desea manifestar un agradecimiento especial al ITESO por brindar la oportunidad de cursar esta maestría y por su constante respaldo académico, al igual que a mi familia por todo el apoyo económico y emocional que me proporcionó.

Asimismo, se desea agradecer a la empresa IZEI Consulting Group, por su apoyo financiero y por otorgar el tiempo necesario para llevar a cabo este trabajo. Su contribución ha sido fundamental para la realización de este proyecto. Por último, el autor desea expresar su gratitud hacia su novia, quien ha brindado un apoyo incondicional durante todo el proceso de la maestría, incluso en los momentos de mayor exigencia y estrés.

Cabe mencionar qué sin el apoyo y colaboración de todas estas personas e instituciones mencionadas, este trabajo no habría sido posible. Su valioso respaldo ha sido crucial en cada etapa de esta investigación, y se les agradece sinceramente su contribución.

DEDICATORIA

Con profunda gratitud y amor, el autor dedica este documento a su madre, Martha Susana Arriola Núñez. Su constante impulso y apoyo incondicional han sido fundamentales en su crecimiento tanto profesional como personal. Desde temprana edad, su madre ha estado a su lado, respaldándolo en cada paso de su carrera.

La confianza inquebrantable y el amor incondicional de su madre han sido una fuente constante de motivación y valentía. Su presencia ha sido una luz en cada momento y lo motiva a alcanzar nuevas metas. Su madre ha sido un ejemplo de fortaleza, perseverancia y dedicación, y su influencia ha sido invaluable.

A través de esta dedicatoria, el autor desea expresar su profunda gratitud y amor hacia su querida madre. Este trabajo de grado es un humilde tributo a su amor incondicional y apoyo constante. Sin ella, el autor no sería la persona que es hoy. Esta dedicatoria es un reconocimiento a su influencia positiva y un recordatorio eterno del impacto que ha tenido en su vida.

RESUMEN

La comprensión del problema de la corrupción es compleja, por lo que necesitamos nuevas herramientas para proponer soluciones innovadoras. Este trabajo se centra en la creación de una de estas herramientas, enfocada en generar escenarios sintéticos para analizar la corrupción.

Con este trabajo se busca facilitar la creación de múltiples escenarios sociales, como familias, escuelas, suburbios y ciudades. Estos escenarios representan diferentes contextos en los que la corrupción puede manifestarse y afectar a la sociedad en general.

Además, se busca proporcionar a los investigadores y analistas una plataforma para comprender mejor los mecanismos de la corrupción y evaluar posibles estrategias para combatirla. Al generar escenarios sintéticos, se pueden realizar análisis exhaustivos y generar experimentos que ayuden a identificar las causas subyacentes y las posibles soluciones para abordar este problema social tan complejo.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ACRÓNIMOS Y ABREVIATURAS

API Application Programming Interface. 29

GDS Graph Data Science. 34

HTML Hypertext Markup Language. 19

HTTP Hypertext Transfer Protocol. 29

IPC Índice de Percepción de la Corrupción. 13, 23

JSON JavaScript Object Notation. 31

PIB Producto Interior Bruto. 22

REST Representational State Transfer. 29, 30

RPaSDT Rumor Propagation and Source Detection Toolkit. 5, 18

UI User Interface. 5, 19

1. INTRODUCCIÓN

Resumen: *En este capítulo se muestran las bases para dar inicio al Trabajo de Obtención de Grado que se realizó a lo largo de la maestría en Sistemas Computacionales por el alumno Oscar René Medina Arriola. El objetivo de este trabajo fue obtener un Modelo basado en grafos para analizar y simular redes de corrupción y observar su comportamiento en el tiempo. En este capítulo se aborda el análisis histórico de la corrupción a lo largo del tiempo, así como los índices utilizados actualmente para medir el nivel de corrupción en una sociedad.*

1.1. Antecedentes

La corrupción es un desafío importante en México que afecta a diversas instituciones, incluyendo las fuerzas policiales. El artículo de los doctores J. O. Gutierrez-Garcia y L.-F. Rodríguez [1] analizan diferentes estrategias para prevenir y combatir la corrupción, tomando como ejemplo la corrupción policial. Se destaca la implementación de la prueba de integridad, la cual se menciona en los siguientes capítulos, como una de las estrategias propuestas [1]. Asimismo, se menciona el Índice de Percepción de la Corrupción, donde México se encuentra entre los países con un alto índice de corrupción [2]. Estos antecedentes nos ayudan a tener contexto base para el desarrollo de este documento.

1.2. Justificación

En las últimas décadas, el desarrollo tecnológico en áreas cómo las telecomunicaciones, la electrónica y la movilidad ha modificado el comportamiento de las personas a tal grado y velocidad que difícilmente se puede estimar el impacto que tendrá en la sociedad y el medio ambiente. A pesar del gran desarrollo tecnológico, son pocos los avances tecnológicos basados en teorías del comportamiento humano que se enfocan en facilitar la comprensión y predicción del impacto que tendrán los cambios del comportamiento humano en la sociedad y su medio ambiente. Por lo anterior, es necesario desarrollar nuevas herramientas que ayuden a estudiar los cambios en el comportamiento humano y su impacto, no solo en fenómenos sociales inmersos en contextos como la migración, la política y corrupción, sino también su impacto en el medio ambiente.

1.3. Problema

De acuerdo con M. A. Casar, la corrupción es el abuso del poder para beneficio propio, esto puede ocurrir en cualquier sector social, ya sea en un ambiente empresarial, político, incluso en un ambiente de soporte caritativo, la corrupción es un problema difícil de detectar, ya que siempre evita ser detectado [2]. Un ejemplo de corrupción en México sería el siguiente: para evitar una multa, algunos ciudadanos eligen dar una cierta cantidad de dinero a la persona que los está infraccionando con la intención de evitar el pago de una multa mayor. Es aquí donde se hace la pregunta, ¿Cuál sería la forma más viable de detectar si un grupo de personas participa en actos de corrupción?

Hoy en día existen algunos métodos para medir y contabilizar estos actos de corrupción, por ejemplo, aplicar encuestas a los mismos ciudadanos, pero de acuerdo a M. A Casar, estos métodos son muchas veces imprecisos e irónicamente estos estudios para medir la corrupción, son corrompidos por los involucrados. Uno de los parámetros más fiables para

medir la corrupción es el IPC (Índice de Percepción de la Corrupción), de acuerdo a sus resultados en el 2019, nos revela que este problema, es un problema global y no se centra solamente en México [2].

En el 2019 de acuerdo al Banco Mundial, México tiene un -0.9 de calificación en un rango de -2.5 a 2.5, siendo -2.5 la más baja posible, y con la posición 16 de 100 en el indicados de control de la corrupción, siendo 100 la mejor posición posible (entre más alta la posición menor es la corrupción), este índice es parecido al IPC y es aplicado cada año entre 209 países, como se puede observar, estos índices son alarmantes y preocupantes, en el siguiente gráfico de acuerdo a los índices calculados en una muestra de 12 países de Latinoamérica en 2019, México se encuentra en los últimos lugares, estando apenas en una mejor posición que Guatemala y Venezuela [2]. (Véase Figura 1.1)

Figura 1.1: Índice de Corrupción en Latinoamerica

Consideramos que el IPC y el índice del Banco de México son pertinentes para tener una valoración global de la corrupción para el país, sin embargo, vemos la oportunidad de proveer herramientas tecnologías que apoyen en el proceso de análisis y permita hacer simulaciones de la forma en que una persona corrupta podría corromper a otro grupo de personas o instituciones.

1.4. Objetivos

1.4.1. Objetivo General

Diseñar e implementar una herramienta para el modelado basado en grafos de redes de corrupción y su comportamiento que facilite su análisis y simulación.

1.4.2. Objetivos Específicos

En el trabajo de obtención de grado se han establecido los siguientes objetivos específicos:

1. Identificar los actores existentes en una sociedad, por ejemplo: personas, instituciones, organizaciones, etc.
2. Proponer una representación formal de los actores en una sociedad, sus atributos (nivel socio-económico, nivel educativo, etc.) y sus relaciones.
3. Desarrollar una herramienta capaz de generar escenarios que ayuden a profesionales sin perfiles técnicos a analizar diferentes grupos de personas que puedan ser corrompidas

1.5. Novedad científica, tecnológica o aportación

En la revisión de la literatura, hemos encontrados software como RPaSDT y el *Web Generator*. Estás herramientas permiten generar escenarios aleatorios en grafos, correr algoritmos y generar datos desde interfaces gráficas, sin embargo, no se conectan directamente con una base de datos basada en grafos, o pueden crear escenarios sintéticos para representar estructuras de una sociedad y simular el proceso de propagación de la corrupción.Debido a estas desventajas, consideramos que nuestra herramienta ofrecerá esa ventaja contra las existentes.

En el siguiente capítulo, se abordará de manera amplia el tema de la corrupción. Se describen en detalle las causas de corrupción, y los índices con los que se ha medido a lo largo del tiempo.

2. ESTADO DEL ARTE O DE LA TÉCNICA

Resumen: *En este capítulo se presenta un análisis de como funciona la corrupción hoy en día, al igual de algunas herramientas que nos ayudan a la generación de grafos y datos.*

2.1. DINÁMICA DE LA CORRUPCIÓN

Actualmente existen varios estudios que han intentado abarcar este problema y de acuerdo con Gambetta (2000) existen 3 agentes involucrados en la corrupción[3]: Donde el primer agente es la persona o colectivo que le ofrece la confianza al segundo agente, el segundo agente es la persona que confía en el primer agente, entre estos dos agentes existen ciertas reglas que ambos deben de cumplir, y existe un tercer agente que busca tentar al segundo agente para romper esta regla, [3] Lo que nos lleva al diagrama definido por Hokky Situngkir (Véase Figura 2.1)

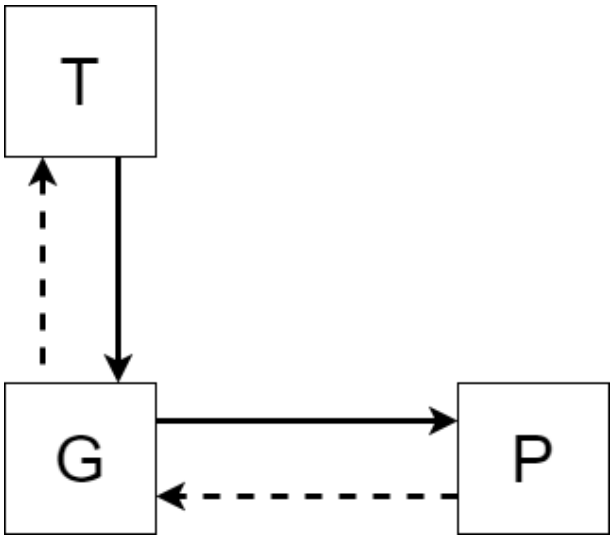

Figura 2.1: Diagrama de Corrupción

En la figura 2.1 se representa al primer agente como T, que tiene una relación con el segundo agente G, donde este segundo agente tiene una relación con el tercer agente P, este ultimo agente P es el que intenta corromper a G, y solo existe una relación entre G y P más no existe la relación con T y P, por esto mismo, la corrupción es un tema difícil de detectar y se puede esparcir muy fácilmente. Utilizando un ejemplo simple, cuando un ciudadano (G) tiene una relación con el gobierno de su ciudad(T) y entre ellos existe una regla de no exceder el limite de velocidad, suponiendo que G rompe esta regla tiene que pagar una multa, entonces el oficial de tránsito (P) detecta a G rompiendo esta regla, lo detiene y lo tienta a pagar una cantidad menor de dinero, donde P saldrá beneficiado, en este caso se cumple el diagrama que nos presenta Situngkir.

Uno de los trabajos que ha estudiado el tema de la corrupción es el titulado [1] *Social determinants of police corruption* llevado a cabo por los Doctores J. Octavio Gutierrez-Garcia y Luis-Felipe Rodríguez en el 2019, en este trabajo se pueden observar distintas estrategias para la prevención de la corrupción dentro de la policía, aquí nos explican que para controlar la corrupción dentro de la policía existe un test de integridad (Prenzler nd Roken 2001), pero de acuerdo al autor McCusker (2006), las estrategias anti-corrupción necesitan una reforma cultural a largo plazo, por lo tanto los autores de este trabajo,

nos presentan un análisis de perfiles socioeconómicos de 103 países mediante árboles de decisión, y en base a este análisis nos proponen principios para ayudar al sector policial a diseñar políticas a largo plazo para evitar la corrupción dentro de este mismo.

De acuerdo a este trabajo se realizaron las siguientes 7 observaciones[1]:

- El nivel de la corrupción policial de un país puede ser caracterizada por reglas condicionales embebidas (if- else) que contienen indicadores para monitorear el desarrollo social.
- El indicador social más alto con respecto a la corrupción policial es el nivel escolar en hombres del país.
- Países con un alto índice de población masculina educada tiene un bajo porcentaje de trabajadores del mismo género que trabajan por cuenta propia. De este modo tienen un bajo porcentaje de corrupción policiaca.
- La mortalidad infantil fue seleccionada como un indicador social relevante al estudio. Este indicador es consecuencia de el indicador del nivel escolar en hombres del país.
- El porcentaje de hombres en el país es relevante para la corrupción en la policía.
- El desarrollo económico y los indicadores de trabajo social son relevantes a la corrupción policial.
- El país de Rwanda es la excepción, ya que este país tiene un bajo índice de educación en hombres y bajo índice de corrupción, esto puede ser gracias que en el año 2003, Rwanda implemento varias reformas para reducir la corrupción policial dentro del país.

Con base en los factores anteriores, la condición para mejorar la corrupción es mejorar los indicadores sociales como la incrementar el nivel de educación en hombres, reducir la fuerza de trabajo informal, incrementar el número de mujeres en el campo laboral y académico, y por último mejorar el desarrollo económico e indiciadores de trabajo social [1].

De acuerdo con los trabajos mencionados en esta sección se puede tener un mayor conocimiento de los factores que pueden afectar ya sea positivamente o negativamente a la corrupción en un país y como esta se puede esparcir, ya que estos trabajos nos proporcionan conceptos, modelos, análisis y causas de la corrupción para generar un modelo que pueda simular redes de corrupción y como pueden evolucionar a lo largo del tiempo.

2.2. Herramientas en la actualidad

A continuación se revisan algunas herramientas que han propuesto a otros autores que contribuyen en la resolución del problema:

2.2.1. RPaSDT: *Rumor Propagation and Source Detection Toolkit*

Esta herramienta permite al usuario llevar a cabo experimentos de propagación de rumores en diferentes topologías de red, utilizando modelos de difusión [4]. Además, basándose en el grafo de propagación, esta herramienta facilita la identificación de posibles fuentes de difusión. Es importante destacar que esta herramienta proporciona una serie de herramientas auxiliares para realizar análisis de redes sofisticados, seleccionar distintas fuentes y verificar el comportamiento de la difusión en una topología y conjunto de fuentes determinados. Es la primera en su categoría que no solo permite simular y visualizar la propagación de información o epidemias, sino que también proporciona un conjunto completo de algoritmos reconocidos para la detección de fuentes en cualquier tipo de red. Además, se trata de un proyecto de código abierto que implementa una amplia variedad de algoritmos de identificación de fuentes de vanguardia y métodos ampliamente conocidos (Véase figura 2.2).

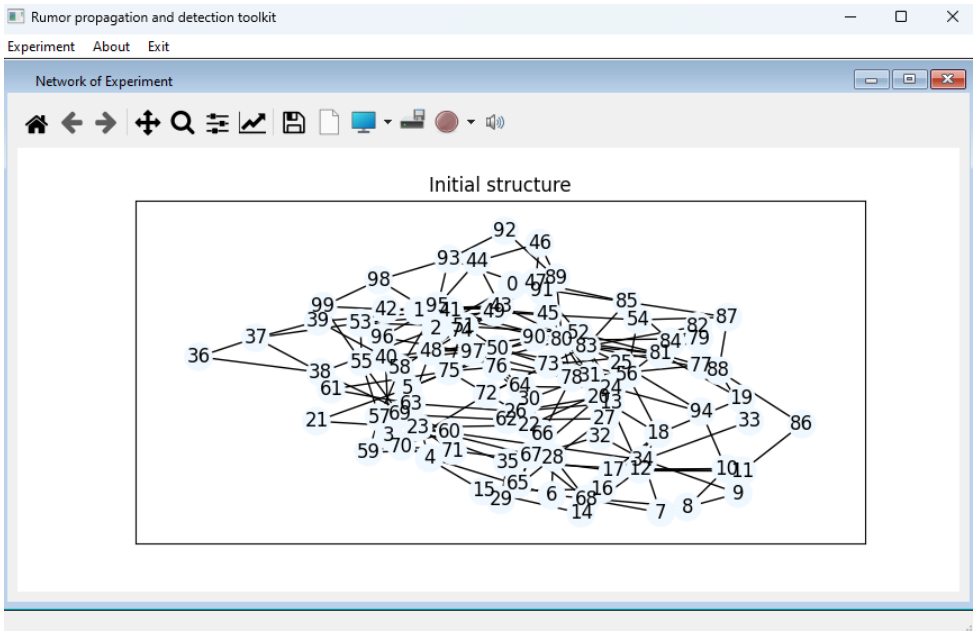

Figura 2.2: Herramienta RPaSDT

2.2.2. Web Generator: UI dataset generation

Es una herramienta llamada *Web Generator* que se utiliza para generar conjuntos de datos de interfaces gráficas de usuario para sitios web. Estos conjuntos de datos incluyen imágenes, descripciones de secciones en formato JSON y código fuente HTML con clases del framework Bootstrap. La herramienta se desarrolló principalmente en Python 3 y Javascript y sigue un enfoque de programación orientada a objetos.

El proceso de generación de datos comienza con la generación de sitios web HTML probabilísticos utilizando probabilidades definidas por el usuario para los componentes y disposiciones de diseño [5]. La herramienta genera archivos HTML, imágenes en formato PNG y archivos JSON con etiquetas para cada elemento web contenido en la imagen. Se utilizan características relacionadas con HTML y se utiliza el controlador Selenium para generar capturas de pantalla. La generación del código HTML se basa en el *framework Bootstrap* para el estilo y la generación de código.

El proceso de generación de datos completo toma las opciones de elementos y diseño, así como las probabilidades como entrada, y produce componentes HTML (las partes más importantes de la página web) y los elementos (los secundarios, generalmente el contenido real). El texto también describe el proceso gráficamente y menciona que se utilizan métodos de selección de elecciones y pesos acumulativos para determinar qué opciones se seleccionan durante la generación. Además, se menciona que se generan capturas de pantalla de las páginas web y se guardan las coordenadas y etiquetas de los elementos en un archivo JSON [5].

En resumen, el *Web Generator* es una herramienta que utiliza Python y Javascript para generar conjuntos de datos de interfaces gráficas de usuario para sitios web, incluyendo imágenes, descripciones de secciones y código fuente HTML [5]. Permite generar sitios web HTML probabilísticos, aplicar estilos con el *framework Bootstrap* y generar capturas de pantalla de las páginas web generadas.

2.2.3. Conclusión

En resumen, los estudios actuales sobre corrupción han permitido comprender mejor sus dinámicas y los actores involucrados, al igual de como la corrupción se encuentra incluso en organizaciones de confianza como es el ejemplo de la policía done indicadores como el nivel educativo, la fuerza laboral informal, la participación de mujeres y el desarrollo económico están relacionados con la corrupción policial.

De acuerdo a las herramientas existentes se identificaron diversas desventajas en relación a estas dos herramientas. Sin embargo, es importante destacar que estas limitaciones no implican que sean herramientas deficientes, sino que no se enfocaron en abordar ciertos aspectos. Estas herramientas no están diseñadas para generar escenarios específicos con atributos en los nodos y relaciones, ni para generar y almacenar grandes volúmenes de

datos en una base de datos, como ocurre en el caso de Neo4J.

Durante la revisión de la literatura, no se ha identificado alguna herramienta que facilite de forma sencilla de modelar la corrupción en distintos escenarios que ayuden a profesionales de áreas como las ciencias sociales. Estos profesionales no necesariamente poseen conocimientos en desarrollo de software.

3. MARCO TEÓRICO

Resumen: *En este capítulo se presentan conceptos de la corrupción donde se definen los términos en los cuales una persona o sociedad puede ser corrompida. Al igual se presentan conceptos acerca de bases de datos basadas en grafos*

3.1. Elementos que inciden en la corrupción

En la actualidad se han encontrado varios factores que influyen en que una sociedad o una persona se corrompa, algunos de estos factores son los siguientes [6]:

- Nivel Educativo
- Imitación/aprendizaje
- Globalización
- Ética
- Prolongación innecesaria y costosa de los trámites
- Creencias y valores culturales
- PIB per cápita
- Moral y normas percibidas
- Sensación de impunidad
- Debilidad de los marcos legales

3.1.1. Factores Educativos y Culturales

La educación es un gran factor en la corrupción, ya que esta se relaciona directamente con lo enseñado por familiares / amigos, lo cual conlleva a seguir el ejemplo de nuestros tutores, ya sean padres o bien maestros, por ejemplo si se ve a nuestro padre dar un soborno a un policía, el hijo o hija puede replicar este comportamiento, y si el nivel educativo se trunca de manera temprana, no hay manera de que el niño pueda distinguir si es una buena o mala práctica hasta que ya es demasiado tarde [6].

La corrupción puede ser aprendida de los familiares / amigos cercanos a la persona. Esto puede ser debido a la presión social o bien a la realidad social en la que viven algunas personas [6].

Desde un punto de valores se ha visto que la corrupción también puede depender directamente del nivel del autoestima de una persona, ya que a una mayor autoestima las probabilidades de que esa persona se corrompa disminuye. También se ha visto que la cultura de un país afecta al nivel de corrupción de este mismo como son los ejemplo de Estados unidos y China, ya que China observa que es mas grave un soborno organizacional que un soborno Individual, a caso contrario de Estados Unidos. Algunos valores culturales causan una pérdida de confianza en el país que viven las personas, esto influye en el nivel

jerárquico de las personas, ya que a mayor nivel jerárquico de una institución / persona, existe un sentimiento de pertenencia sobre los recursos públicos, al igual que las personas o instituciones mejores observan una falta de compromiso con las instituciones. [6].

3.1.2. Globalización y Factores Gubernamentales

La corrupción como causa de la globalización se debe principalmente a una mayor circulación de capitales y la falta de medios legales y políticos a nivel global [6], ya que países externos no pueden controlar la corrupción de otro país que se rige a leyes distintas [6].

Existen actualmente varios procesos burocráticos en nuestro país que propensa a la corrupción a causa de una prolongación de estos trámites [6], algunos de estos ejemplos son:

- Agilizar trámites.
- Obtener licencias y permisos.
- Abuso de autoridad.
- Generar contratos.
- Participación en licitaciones.

También se observa que a un mayor PIB per cápita existe menor riesgo de corrupción, o dicho de otra manera un mayor IPC.

3.1.3. Factores Morales y Éticos

La ética influye en la corrupción gracias a la toma de decisiones de individuos u organizaciones,esto se debe a que una orientación empresarial causa un mayor índice de corrupción que una orientación a la innovación [6].

De acuerdo a la moral y normas percibidas, cuando alguien con mas poder o un poco con mas de conocimiento acerca de las leyes y reglas, es mas probable que abuse de ese poder y cometa actos de corrupción [6]. Estas normas puede llevar a algunas personas a una sensación de impunidad, este punto es de los causantes mas importantes, ya que si alguien se siente por encima de la ley, o bien que es intocable, esa persona es mas probable a corromperse [6], esto nos lleva a la debilidad y existencia de los huecos en los marcos legales, ya que estos son causantes de que las personas se aprovechen de ellos y utilicen esos puntos débiles de las leyes para corromper o realizar actos de corrupción y salir impunes [6].

3.1.4. Conclusión

En resumen, la corrupción se ve influenciada por diversos factores, comprender estos factores es fundamental para abordar eficazmente el problema de la corrupción y promover sociedades más justas y transparentes. Es necesario fortalecer estos factores o causas para en un futuro poder prevenir la corrupción en la mayor medida posible.

3.2. Bases de datos orientada a grafos

3.2.1. Definición

Una base de datos orientada a grafos es una plataforma especializada que se utiliza para crear y manipular grafos. Los grafos son estructuras que constan de nodos, bordes y propiedades, y se utilizan para representar y almacenar datos de una manera diferente a las bases de datos relacionales.

La analítica de grafos se refiere al proceso de analizar datos que están estructurados en forma de grafo. En este tipo de análisis, los datos se representan como nodos, mientras que las relaciones entre ellos se representan como bordes. Para llevar a cabo análisis de grafos, es necesario utilizar una base de datos que admita este formato, como una base de datos especializada en grafos o una base de datos convergente que soporte múltiples modelos de datos, incluyendo los grafos. Una base de datos orientada a grafos es una plataforma especializada y de un solo propósito para crear y manipular grafos. Los grafos contienen nodos, bordes y propiedades que se utilizan para representar y almacenar datos de una forma que no permiten las bases de datos relacionales [7].

3.2.2. Nodos y Relaciones

Los nodos son la entidad que se distingue inicialmente en un grafo [6]. Estos nodos se pueden agrupar en diferentes tipos de datos y tener distintas propiedades, estas propiedades son llamados como atributos, de esta manera se facilita la descripción de cada nodo, y ayuda a poder consultarlos en una base de datos, no hay un límite para el número de atributos [8].

Para las relaciones se identifican las interacciones que hay entre los nodos [8], como se muestra en la figura 3.1, donde el nodo A tiene una relación hacia el nodo B de amistad, y de manera viceversa, lo que nos lleva a que un grafo puede ser dirigido o no dirigido.

Un grafo dirigido nos indica que dentro de un grafo las relaciones tienen un nodo origen y un nodo destino[8], como es el ejemplo de la figura 3.2, donde se nos indica que el Nodo

A es jefe del Nodo B. Al igual que los nodos, las relaciones pueden tener atributos, en este ejemplo podríamos colocar el tiempo en el que el nodo A ha sido jefe del nodo B.

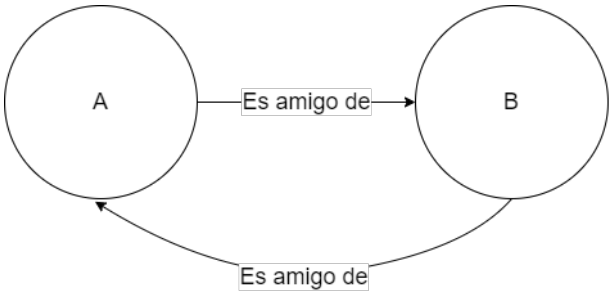

Figura 3.1: Ejemplo de Grafo simple con relación de amistad

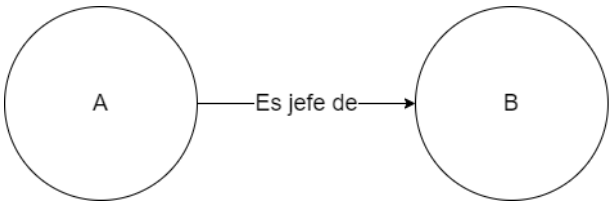

Figura 3.2: Ejemplo de Grafo simple con relación de jefatura

3.2.3. Ejemplos de bases de datos basadas en grafos

Actualmente existen varias bases de datos basadas en grafos, cabe aclarar que la tecnología que usaremos nosotros es Neo4J pero algunas de las más reconocidas son las siguientes:

- Neo4J: Está diseñada específicamente para almacenar, administrar y consultar datos en forma de grafos. Utiliza un modelo de datos de grafo nativo que permite representar relaciones complejas entre entidades y realizar consultas eficientes en el grafo y un lenguaje de consulta llamado Cypher. Neo4j es ampliamente utilizado en aplicaciones que requieren un alto rendimiento en consultas de grafos, como redes sociales, análisis de redes, recomendaciones personalizadas y detección de fraudes [9].
- Amazon Neptune: Está diseñado para almacenar y procesar grandes conjuntos de datos en forma de grafos. Neptune es compatible con el lenguaje de consulta de grafos de código abierto llamado Apache TinkerPop Gremlin, lo que facilita el desarrollo de aplicaciones basadas en grafos. Es altamente escalable y se integra bien con otros servicios de AWS, lo que lo hace adecuado para aplicaciones en la nube que requieren un almacenamiento y procesamiento eficiente de datos en forma de grafos [10].
- JanusGraph: Está diseñada para manejar grandes volúmenes de datos y consultas complejas en entornos distribuidos. JanusGraph utiliza un modelo de datos de grafo etiquetado que permite representar múltiples tipos de relaciones entre los nodos. Se basa en Apache Cassandra y Apache HBase para el almacenamiento subyacente, y al igual que Amazon Neptune utiliza Apache TinkerPop Gremlin como lenguaje de consulta. [11]

3.2.4. Algoritmos

Los algoritmos en las bases de datos basadas en grafos son técnicas computacionales diseñadas específicamente para operar y analizar datos almacenados en forma de grafo. Estos algoritmos permiten realizar diversas operaciones y consultas eficientes en la estructura de grafo, como encontrar caminos más cortos, calcular la importancia de los nodos, detectar comunidades y descubrir patrones estructurales. Algunos de estos algoritmos son los siguientes:

- Page Rank: Mide la importancia de cada página dentro de un conjunto de páginas web, basándose en la cantidad de enlaces que recibe y en la importancia de las páginas que le enlazan. La idea principal es que una página es importante si otras páginas importantes la enlazan. En resumen, el algoritmo PageRank ayuda a determinar qué páginas son más relevantes en función de los enlaces que reciben y la importancia de las páginas que las enlazan [12].

- Degree Centrality: Se utiliza para encontrar nodos populares dentro de un grafo. La centralidad de grado mide el número de relaciones entrantes o salientes (o ambas) desde un nodo, dependiendo de la orientación de una proyección de relaciones. Se puede aplicar tanto a grafos con pesos como sin pesos [13].

- Betweenness Centrality: Se utiliza para identificar nodos que actúan como puentes entre diferentes partes del grafo. El algoritmo calcula las rutas más cortas entre todos los pares de nodos en el grafo y asigna a cada nodo una puntuación basada en la cantidad de rutas que pasan a través de él. Los nodos que se encuentran con mayor frecuencia en las rutas más cortas entre otros nodos tienen una mayor centralidad de intermediación. El algoritmo puede aplicarse a grafos sin pesos o con pesos positivos, utilizando diferentes técnicas de cálculo [14].

- Louvain: Se utiliza para detectar comunidades en redes grandes. Se enfoca en maximizar un puntaje de modularidad para cada comunidad, lo que implica evaluar qué tan conectados están los nodos dentro de una comunidad en comparación con una red aleatoria. El algoritmo utiliza un enfoque de agrupamiento jerárquico, fusionando de manera recursiva comunidades en un solo nodo y ejecutando el agrupamiento de modularidad en los grafos condensados. Esto permite identificar estructuras comunitarias dentro de la red [15].

- Label Propagation: Utiliza la estructura de la red para guiar la detección de comunidades, sin requerir una función objetivo predefinida o información previa sobre las comunidades. El algoritmo propaga etiquetas a través de la red y forma comunidades en función de este proceso. La idea central es que una etiqueta puede volverse dominante en un grupo densamente conectado de nodos, pero tendrá dificultades para cruzar regiones escasamente conectadas. Al final del proceso, los nodos con la misma etiqueta se consideran parte de la misma comunidad [16].

- Weakly Connected Components: Se utiliza para encontrar conjuntos de nodos conectados en grafos dirigidos y no dirigidos. Se considera que dos nodos están conectados si existe un camino entre ellos, sin importar la dirección de las relaciones en ese camino. Los nodos que están conectados entre sí forman un componente. Se utiliza comúnmente al inicio de un análisis para comprender la estructura global del grafo y permite ejecutar otros algoritmos de manera independiente en los conjuntos de nodos identificados como componentes débilmente conectados [17].

4. DESARROLLO METODOLÓGICO

Resumen: *Este capítulo tiene como propósito presentar de manera detallada la metodología empleada para desarrollar la herramienta de este trabajo de grado. En esta sección se describirán los pasos seguidos,desde la investigación inicial sobre Neo4j, hasta la implementación. Se explicarán los métodos utilizados para establecer la conexión entre el cliente y la base de datos. Este capítulo ayuda a una comprensión clara y precisa de cómo se logró la integración exitosa entre el cliente y la base de datos de Neo4j.*

4.1. Investigación sobre Neo4J

La primera etapa consistió en realizar una investigación sobre Neo4J, la base de datos basad en grafos que me fue asignada como herramienta para trabajar. Se exploraron sus características principales, su arquitectura y su lenguaje de consulta Cypher. Se investigó cómo Neo4J almacena y gestiona datos relacionales en forma de grafos, lo que permite modelar y analizar relaciones complejas de manera eficiente. Se examinaron casos de uso y ejemplos de aplicaciones exitosas que utilizan Neo4J como motor de base de datos.

Esta investigación proporcionó una comprensión sólida de los fundamentos de Neo4J y su facilidad para resolver problemas complejos que involucran relaciones. Además, ayudó a comprender de manera efectiva en el contexto del desarrollo de la herramienta.

4.2. Investigación sobre REST API para conectar Neo4J con JavaScript

En esta sección se centró en investigar las mejores prácticas para conectar Neo4J con JavaScript utilizando REST API. Se estudiaron las diferentes formas de interactuar con la base de datos Neo4J a través de peticiones HTTP y se analizaron las ventajas y desventajas de utilizar este enfoque. Se investigaron bibliotecas para facilitar la comunicación entre JavaScript y Neo4J, y se decidió utilizar "neo4j-driver". Se examinaron ejemplos de código y se evaluaron diferentes formas para realizar consultas y actualizaciones en Neo4J desde el lado del cliente. Se investigó lo necesario para el manejo de autenticación y autorización en la conexión entre Neo4J y JavaScript. Todo lo anterior, permitió una integración de Neo4J con el cliente de javascript, y aseguró una comunicación segura y eficiente entre el cliente y la base de datos.

4.3. Diseño de la herramienta

En esta sección, se llevó a cabo el diseño de la herramienta que se iba a desarrollar. Se definieron los requisitos funcionales y no funcionales, considerando las necesidades específicas de los usuarios y los objetivos del proyecto. Se recopiló la información obtenida durante las etapas de investigación sobre Neo4J y REST API para facilitar el diseño de la herramienta.

Se elaboró un diagrama de arquitectura como se muestra en la figura 4.1, que representa la interacción entre los diferentes componentes de la herramienta. Se identificaron las funcionalidades clave que la herramienta debía ofrecer, como la visualización de datos en forma de gráficos, la capacidad de realizar consultas y actualizaciones en la base de datos

Neo4J.

4.3.1. Arquitectura

La arquitectura de la solución propuesta se compone de una serie de elementos que trabajan en conjunto para lograr un funcionamiento eficiente y efectivo. A continuación, se detallan los puntos clave de esta arquitectura, resaltando su importancia y las interacciones entre ellos como se muestra en la figura 4.1

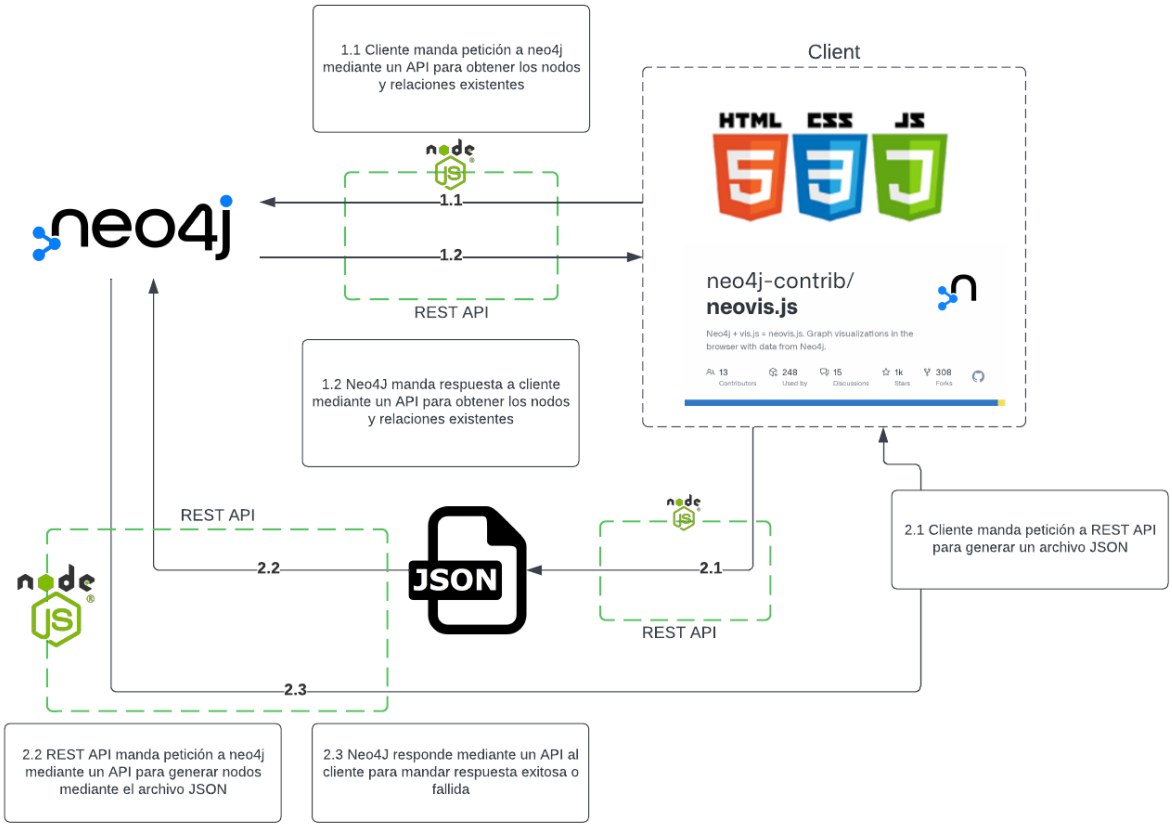

Figura 4.1: Diagrama de Arquitectura

- **Consulta de datos y ejecución de algoritmos en Neo4j:** En la figura 4.1 en la sección 1.1 del flujo de trabajo, el cliente inicia una petición REST mediante Node.js hacia la base de datos de Neo4j. Esta petición tiene como objetivo obtener los nodos y relaciones almacenados en la base de datos. Además, se pueden incluir sentencias para ejecutar algoritmos específicos directamente en Neo4j. Esto permite acceder y manipular los datos de manera eficiente.
- **Respuesta de Neo4j y entrega de resultados:** Una vez recibida la petición en Neo4j, se procede a procesarla y generar una respuesta en la figura 4.1 sección

1.2 del flujo de arquitectura. Neo4j responde a Node.js proporcionando los nodos y relaciones existentes en la base de datos. Esta respuesta se envía en un formato estructurado y legible, lo que facilita su procesamiento posterior. En caso de haber solicitado la ejecución de algún algoritmo, Neo4j también incluirá los resultados generados por dicho algoritmo. Esta capacidad de ejecutar algoritmos en Neo4j amplía las posibilidades de análisis y manipulación de los datos, brindando al cliente una amplia gama de herramientas y funcionalidades.

- **Visualización de nodos y relaciones con NeoVis y resultados de algoritmos:** Una vez que el cliente ha recibido la respuesta de Neo4j, se visualizar los nodos y relaciones obtenidos utilizando la biblioteca NeoVis. Esta herramienta permite representar visualmente los datos almacenados en Neo4j de manera intuitiva y comprensible. Al interactuar con NeoVis. Si se han ejecutado algoritmos específicos, el cliente también puede mostrar los resultados obtenidos en un formato de tabla.

- **Generación de archivos JSON para estructuras masivas:** Cuando se generan estructuras de datos masivas con un gran número de nodos y relaciones, puede resultar ineficiente manejar toda esta información en tiempo real. Por lo tanto, en el punto 2.1 del flujo de trabajo, se realiza una petición al *backend* para generar un archivo JSON. Este archivo se utiliza para almacenar de manera eficiente la información relacionada con estas estructuras complejas. Al utilizar un archivo JSON, se garantiza una representación compacta y fácilmente transportable de los datos, lo que facilita su almacenamiento y manipulación.

- **Envío del archivo JSON a Neo4j:** Una vez que el archivo JSON ha sido generado, en la figura 4.1 el punto 2.2 del flujo de arquitectura, este se envía a Neo4j mediante Node.js. Neo4j procesa el archivo JSON y lo almacena en la base de datos. Esta capacidad de importar y exportar datos utilizando archivos JSON es crucial en entornos donde se requiere un manejo eficiente de grandes volúmenes de información.

4.4. Desarrollo de código usando Javascript

El desarrollo de JavaScript se dividió en dos partes principales: primero REST API utilizando Node.js y segundo el cliente utilizando JavaScript.

4.4.1. REST API con Node.js

En esta etapa, se utilizó Node.js para desarrollar la REST API que actuaría como intermediario entre la herramienta y la base de datos Neo4J. Se configuraron las rutas y los controladores para manejar las peticiones HTTP entrantes y se implementaron las operaciones necesarias para consultar y actualizar la base de datos Neo4J. Se utilizaron las bibliotecas y *frameworks* investigados anteriormente, como "neo4j-driver", para establecer

la conexión con la base de datos y ejecutar las consultas Cypher necesarias. Se realizaron pruebas exhaustivas para asegurarse de que la REST API funcionara correctamente y respondiera de manera eficiente a las solicitudes.

4.4.2. Desarrollo del cliente usando JavaScript

En esta etapa, se desarrolló el cliente de la herramienta utilizando JavaScript. Se utilizaron las capacidades de NeoVis para visualizar los datos almacenados en Neo4J en forma de gráficos interactivos. Se implementaron las funciones necesarias para comunicarse con la REST API y obtener los datos requeridos. Se diseñó una interfaz de usuario que permitiera a los usuarios interactuar con los gráficos y realizar consultas específicas en la base de datos. Se realizaron pruebas y ajustes constantes para garantizar un rendimiento óptimo y una experiencia fluida para el usuario.

Al igual que se realizo el ensamble de los parámetros de configuración para insertar los distintos escenarios del usuario final para que el *backend* pudiera escribir el archivo JSON y mandarlo a Neo4j.

4.4.3. Funcionalidades de la herramienta

La herramienta ofrece una variedad de funcionalidades que permiten la creación y manipulación de nodos y relaciones en una base de datos Neo4j. Estas funcionalidades se diseñaron con el objetivo de facilitar la creación de estructuras, nodos y relaciones al igual que la construcción de estructuras masivas en la base de datos. A continuación, se describen en detalle las principales funcionalidades de la herramienta:

- **Creación de un solo nodo.** La herramienta proporciona la capacidad de generar un solo nodo en la base de datos de Neo4j. Esta funcionalidad resulta útil cuando se desea agregar un elemento individual a la base de datos. Por ejemplo, crear un nodo de tipo persona con los atributos nombre, edad y ocupación. Esto permite una gestión más granular y detallada de los datos en la base de datos.
- **Relación 1-1 de nodos.** La herramienta permite establecer relaciones directas y únicas entre nodos en un formato de uno a uno. Esto significa que se puede crear una relación específica entre dos nodos en la base de datos. Por ejemplo, crear una relación amigo de entre dos nodos de tipo persona. Al especificar los atributos asociados a esta relación, se pueden agregar detalles adicionales, como la fecha de inicio de la amistad. Como se muestran en las figuras 7.7 y 7.8
- **Creación de estructuras masivas.** La herramienta brinda la facilidad de generar estructuras masivas con múltiples nodos y relaciones. Esta funcionalidad es especialmente útil cuando se requiere crear un gran número de instancias de una

misma estructura. Por ejemplo, crear 100 salones de clases, cada uno con sus nodos y relaciones correspondientes, uno de los ejemplos se puede visualizar en la imagen 7.18

Dentro de esta funcionalidad de creación de estructuras masivas, se incluyen características adicionales para facilitar la manipulación eficiente de los datos:

- **Creación de varios nodos simultáneamente.** La herramienta permite generar múltiples nodos simultáneamente. Esta capacidad de generación facilita el proceso de creación de nodos repetitivos. Esto es especialmente útil cuando se necesita crear una gran cantidad de nodos del mismo tipo en la base de datos. Por ejemplo, crear 20 nodos de tipo empleado en una organización, cada uno con los atributos nombre, puesto y salario.
- **Creación de relación uno a todos los nodos.** La herramienta permite establecer relaciones entre un nodo origen y todos los demás nodos en una misma estructura. Por ejemplo, en un contexto educativo, se puede crear una relación en la cual un nodo tipo maestro se relaciona con varios nodos tipo alumno, indicando que el maestro enseña a varios alumnos. Al establecer esta relación uno a muchos, se facilita el seguimiento y la gestión de las relaciones y permite una estructuración clara de la información en la base de datos.
- **Creación de relación nodo a nodo.** La herramienta permite generar relaciones de uno a uno dentro de cada estructura. Esto significa que se pueden establecer relaciones personalizadas entre nodos individuales dentro de la misma estructura. Por ejemplo, en un contexto escolar, se puede establecer una relación de tipo amistad entre dos nodos alumno dentro de un salón de clases. De esta manera se permite modelar interacciones sociales y conexiones específicas entre los alumnos.

  Estos tipos de relaciones se pueden seleccionar en la pantalla de relaciones dentro de la funcionalidad de generar estructuras, como se muestra en la imagen 7.14

Estas funcionalidades proporcionan a los usuarios una gran flexibilidad y eficiencia para crear y manipular nodos y relaciones en la base de datos Neo4j. La herramienta se adapta a una variedad de escenarios y necesidades, lo que permite la construcción ágil y la gestión efectiva de estructuras complejas en la base de datos.

4.5. Pruebas y ajustes para inserción de grafos masivos

En esta etapa, se realizó un enfoque especial en probar y ajustar la herramienta para manejar la inserción de grafos masivos en la base de datos Neo4J. Se generaron conjuntos de datos de prueba con miles de nodos y relaciones para simular escenarios de alto volumen de datos. Estas pruebas ayudaron a evaluar el rendimiento de la herramienta y

su capacidad para manejar grandes volúmenes de datos de manera eficiente. Durante las pruebas, se monitorizó de cerca el rendimiento de la herramienta, incluyendo la velocidad de inserción de datos, el tiempo de respuesta de las consultas y el consumo de recursos del sistema. Se realizaron ajustes en la implementación para optimizar el rendimiento y minimizar el impacto en el rendimiento durante la inserción de grafos masivos. Se optimizaron las consultas utilizando índices y estrategias de optimización de consultas específicas de Neo4J. Al final se optó por la utilización de librerías como "fs" para escribir un archivo json mediante un input con los parámetros para construir este json, una vez generado el archivo se manda una petición a Neo4J para insertar estos nodos y relaciones en la base de datos, cabe aclarar que ambas funciones son asíncronas ya que es un proceso pesado cuando se tienen escenarios muy grandes.

4.6. Implementación de algoritmos básicos de Neo4J en el cliente de JavaScript

En esta etapa, se implementaron algoritmos básicos de análisis de grafos, que ya están implementados en la librería GDS de Neo4J, en el cliente de JavaScript para realizar análisis y consultas avanzadas en la base de datos. Los algoritmos de comunidad y de centralidad, se integraron en la herramienta.

La implementación de estos algoritmos se basó en la documentación y ejemplos proporcionados por Neo4J. Se investigó cómo utilizar estos algoritmos en el contexto del cliente de JavaScript y se adaptaron para cumplir con los requisitos específicos de la herramienta. Los resultados de los algoritmos se presentaron de manera visual y se integraron con la funcionalidad de visualización de gráficos proporcionada por NeoVis. De manera que los usuarios puedan comprender mejor los datos almacenados en Neo4J y realizar análisis en base a estos resultados.

5. RESULTADOS Y DISCUSIÓN

Resumen: *En este capítulo se muestran los resultados obtenidos durante la maestría, demostrando las capacidades y funcionalidades de la herramienta desarrollada, así como diversos ejemplos de escenarios que pueden ser representados dentro de la herramienta.*

5.1. Escenarios

A continuación se mostraran algunos resultados de escenarios que se utilizó en la herramienta, dentro de los escenarios se muestran imágenes tomadas desde neo4j, ya que la herramienta no muestra todos los nodos y relaciones cuando el escenario es demasiado grande, la interfaz del usuario de la página principal la podrán encontrar en la figura 7.1 dentro de los anexos, cualquier paso sobre los procesos de creación también se encuentran dentro de los anexos.

5.1.1. PROMOCIÓN LABORAL DEBIDO A RELACIÓN SENTIMENTAL

En el escenario representado en la figura 5.1, se observan dos nodos de tipo "persona". Los atributos de cada nodo se detallan en las figuras 5.2 y 5.3. Este escenario ilustra las relaciones existentes entre un gerente de área y un practicante, donde el primero se aprovecha de su posición de poder sobre la segunda persona. A su vez, el practicante experimenta sentimientos hacia el gerente debido a esta dinámica de poder, lo que lleva al ofrecimiento de un ascenso por parte del gerente al nodo "Practicante".

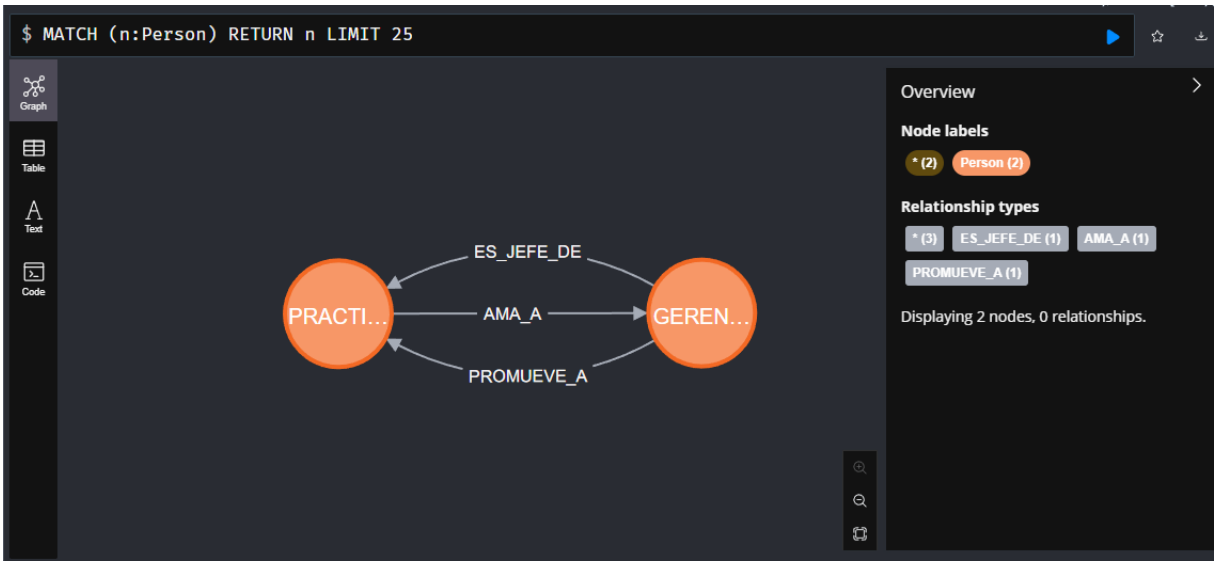

Figura 5.1: Escenario Dinámicas de Poder

Figura 5.2: Atributos del Nodo Practicante

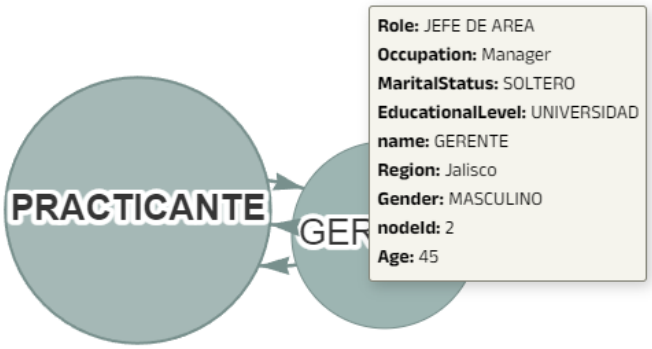

Figura 5.3: Atributos del Nodo Gerente

5.1.2. SOBORNO A OFICIAL DE TRÁNSITO

Este escenario nos muestra dos situaciones que puede ocurrir en el día a día de cualquier ciudadano, para generar este escenario se crearon los nodos persona (un oficial de tránsito y un conductor) con los atributos que se muestran en las figuras 5.4 y 5.5, las dos situaciones de corrupción son las siguientes:

- Situación A: Como se muestra en la figura 5.6, el conductor ofrece un soborno al oficial de tránsito de 2000 pesos mexicanos, el cual el oficial de tránsito acepta.
- Situación B: Como se muestra en la figura 5.7, el oficial solicita un soborno al oficial de tránsito de 5000 pesos mexicanos, el cual el conductor accede a pagar.

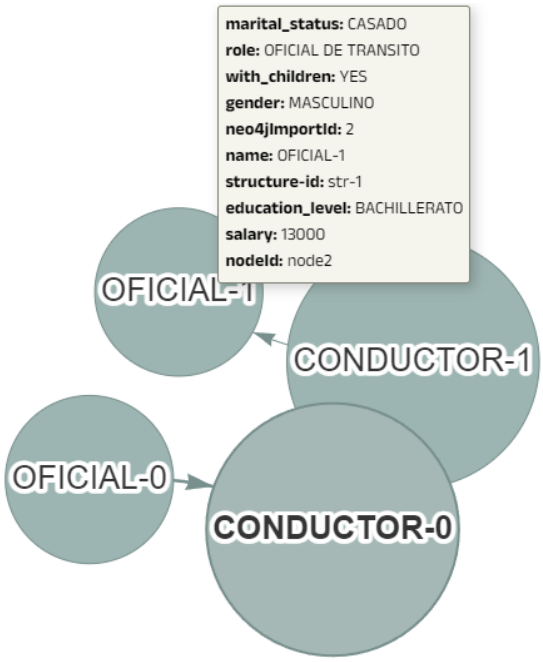

Figura 5.4: Atributos del Oficial de Transito

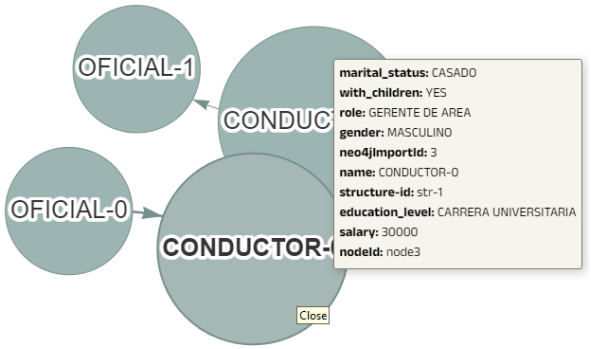

Figura 5.5: Atributos del Conductor

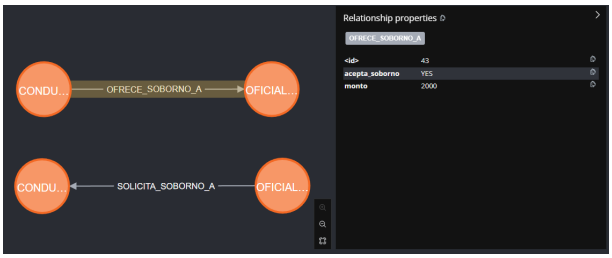

Figura 5.6: Situación A

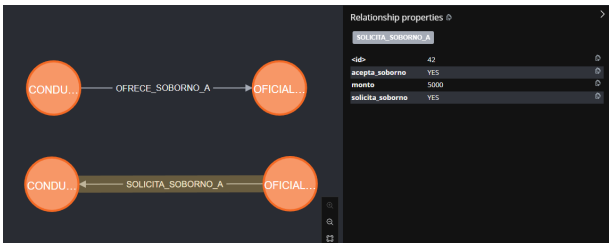

Figura 5.7: Situación B

5.1.3. CREACIÓN DE UN EQUIPO DE TRABAJO DENTRO DE UNA COMPAÑÍA

En este escenario, se utilizó una herramienta para generar un contexto que puede ser corrompido, tal como se muestra en la figura 5.8. Este contexto representa a varios equipos de trabajo dentro de una compañía, cabe señalar que estas imágenes son de neo4j, pero el escenario se creo con la herramienta desarrollada en la maestría, como se menciono anteriormente, este escenario es un poco mas complejo que los anteriores por lo tanto no se logra visualizar completo en la herramienta debido a cuestiones de rendimiento:

Equipo de *Marketing*: Se crearon tres equipos de Mercadotecnia, cada equipo tiene siete licenciados en mercadotecnia como se muestra en la figura 5.9

Equipo de Operaciones: Se generó un ejemplo que se presenta en la figura 5.10, el cual representa un equipo de operaciones. Este equipo está liderado por un Gerente, quien supervisa a un grupo de Desarrolladores. A su vez, el equipo de Desarrolladores está compuesto por un *Senior Engineer*, quien tiene bajo su responsabilidad a tres desarrolladores especializados en una tecnología denominada *Salesforce*.

Equipo de Diseño: Se genero una estructura de un equipo de diseño el cual consta de un Diseñador a cargo de cuatro diseñadores con una experiencia menor como se muestra en la figura 5.11

Equipo de Proyecto: Finalmente se representa un equipo que trabaja en un proyecto dentro de la compañía, el cual tiene como objetivo generar tres integraciones complejas, donde la plataforma llamada *"Twilio"* mandará información a la plataforma *"Salesforce"*, este equipo se muestra en la figura 5.12, este equipo esta conformado por un *Project Manager* el cual gestiona a todo el equipo, pero dentro de este equipo tenemos a un Líder de Desarrolladores especializados en *Twilio*, el cual tiene a su cargo a tres desarrolladores enfocados en esta tecnología, y finalmente también existe un Desarrollador Senior enfocado a *Salesforce* el cuál se encargará del desarrollo del proyecto dentro de la plataforma.

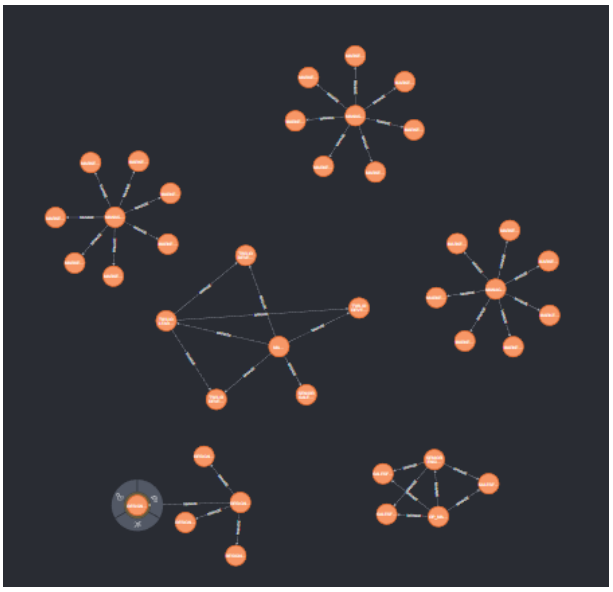

Figura 5.8: Equipos de trabajo dentro de una Compañía de Desarrollo

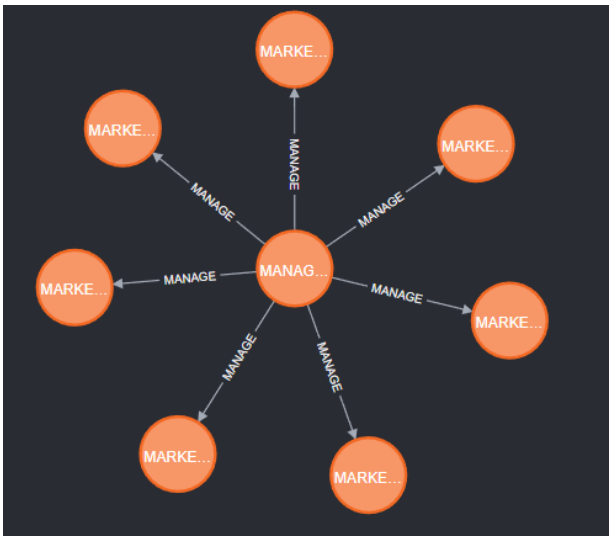

Figura 5.9: Equipo de Mercadotecnia dentro de una Compañía de Desarrollo

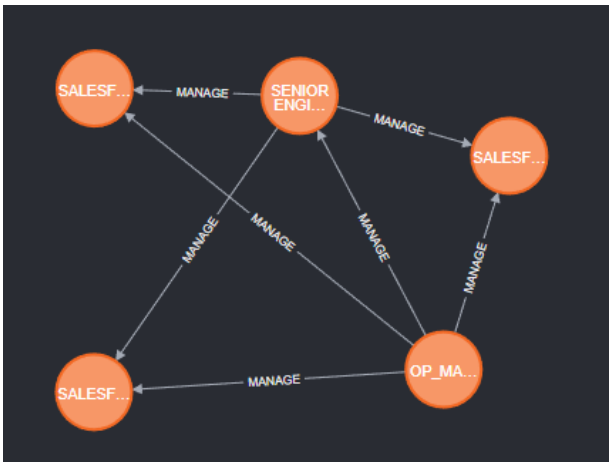

Figura 5.10: Equipo de Operaciones dentro de una Compañía de Desarrollo

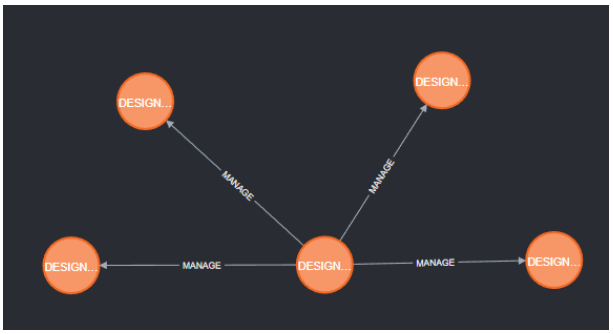

Figura 5.11: Equipo de Diseño dentro de una Compañía de Desarrollo

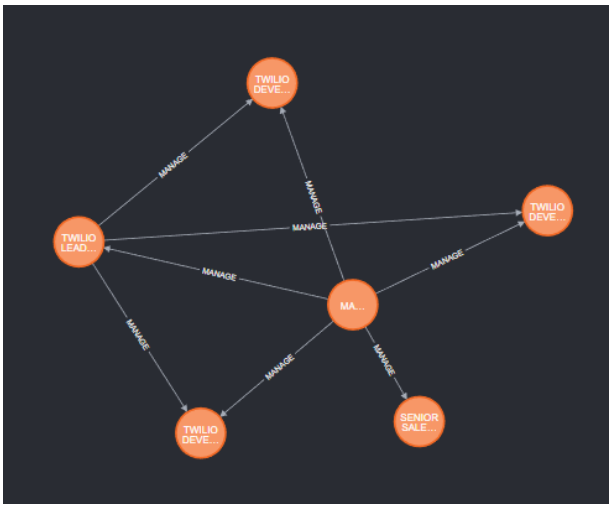

Figura 5.12: Equipo de Proyecto dentro de una Compañía de Desarrollo

5.2. Discusión

Esta herramienta proporciona una gran ventaja en la generación de escenarios masivos gracias al poder de Neo4j. A diferencia de las herramientas previamente mencionadas en capítulos anteriores, como es el ejemplo que se muestra en la figura 2.2 la herramienta actual tiene una conexión directa con Neo4j, lo que permite la generación de escenarios mucho más grandes y la capacidad de agregar atributos a los nodos y relaciones.

La herramienta anterior no era capaz de generar escenarios de gran tamaño y carecía de la capacidad de agregar atributos. Aunque visualmente era más sencilla de usar para los usuarios con un perfil técnico, no guardaba los grafos en una base de datos, lo que implicaba que cualquier trabajo realizado se perdía al cerrar el programa. En mi opinión, ambas herramientas son excelentes, pero cumplen propósitos diferentes. Los escenarios generados por esta nueva herramienta son un poco más aleatorios, ya que no se pueden definir manualmente las relaciones entre los nodos.

En resumen, si necesitas generar escenarios específicos para abordar problemáticas como la corrupción, la herramienta desarrollada en este Trabajo de Grado puede facilitar enormemente esta tarea ya que aprovecha el poder que tiene Neo4J y estos escenarios son guardados en su base de datos. Por otro lado, si lo que buscas es ejecutar algoritmos en grafos grandes sin atributos y escenarios generados por computadora sin contexto específico, la herramienta RPaSDT puede ser de gran ayuda. De esta manera, puedes visualizar varios algoritmos y observar cómo se comportan en diferentes situaciones.

6. CONCLUSIONES

Resumen: *En este capítulo se presentan las conclusiones y trabajo futuro.*

6.1. Conclusiones

En conclusión, este trabajo de grado ha identificado a las personas dentro de una sociedad que pueden influir en la propagación de la corrupción. Esto ha permitido generar diversos escenarios masivos, como un salón de clases o incluso múltiples salones. Aunque no se ha desarrollado un índice específico para evaluar la posibilidad de corrupción de un nodo, se ha investigado sobre el Índice de Percepción de la Corrupción (IPC), que puede ser fácilmente incorporado como atributo en los nodos de la herramienta.

En relación a la generación de escenarios, se han propuesto nodos de tipo persona, como el nodo de empleado, que ya cuentan con atributos definidos y pueden ser fácilmente modificados según sea necesario.

También se destaca que aún falta implementar algoritmos de propagación de la corrupción. Sin embargo, en la actualidad, la herramienta es capaz de generar escenarios simples y masivos, lo que en el futuro permitirá facilitar y ahorrar tiempo en la generación de escenarios donde se simule esta propagación.

Para finalizar se cumplieron los siguientes objetivos:

1. Se identificaron actores y estructuras dentro de una sociedad. Ambos pueden ser representados dentro de la herramienta.
2. Se propuso una forma de representar actores, estructuras, relaciones y sus respectivos atributos dentro de la herramienta.
3. Se desarrolló una herramienta capaz de generar escenarios simples y masivos.

6.2. Trabajo Futuro

En el futuro, se espera mejorar la herramienta para simular la propagación de la corrupción. Esto se logrará mediante la incorporación de agentes, lo que permitirá considerar a diferentes actores y su interacción en entornos corruptos. De esta manera se podrán agregar modelos previamente entrenados con *deep learning* para observar la propagación de la corrupción.

Otro aspecto importante será la expansión de los nodos en la herramienta. Actualmente, se limitan a representar actores y estructuras, sin embargo, se planea agregar macro-nodos que reflejen unidades sociales más grandes, como sociedades o comunidades relacionadas con otras. Esto permitirá comprender cómo los patrones de comportamiento y las dinámicas colectivas influyen en la propagación de la corrupción.

7. APÉNDICE

7.1. Creación de un único nodo

En este escenario, se realiza la creación de un solo nodo siguiendo los siguientes pasos

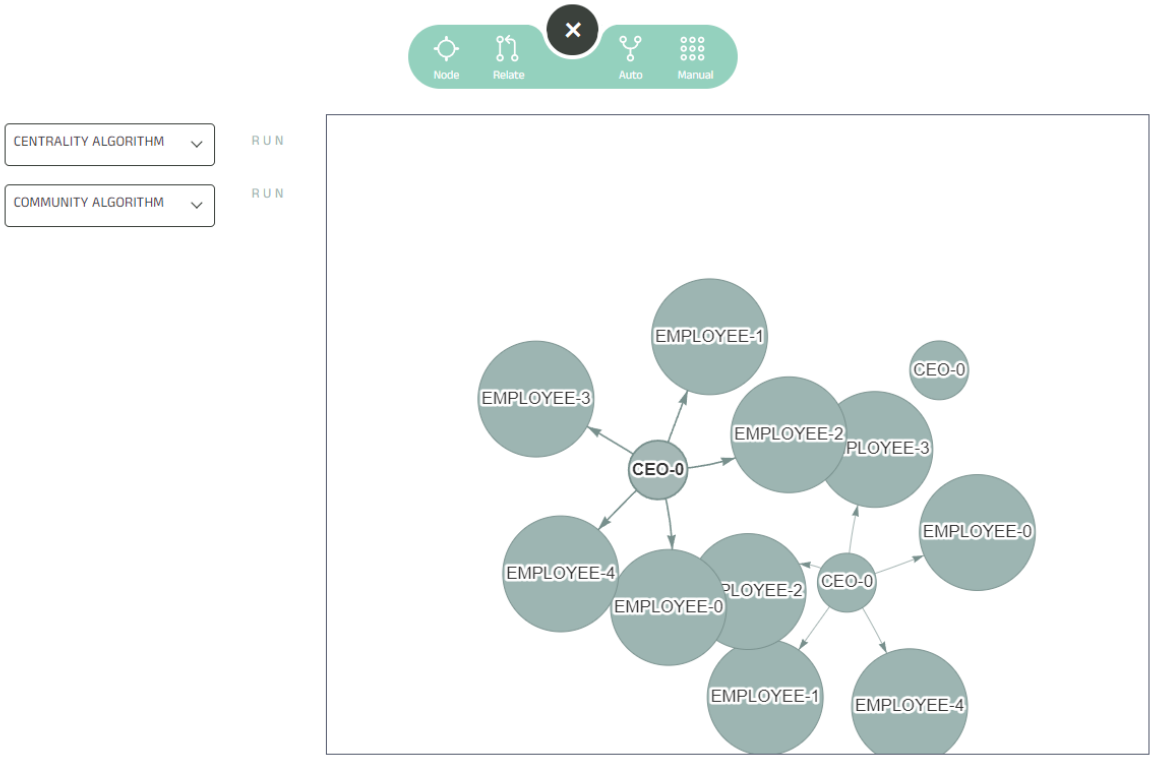

Figura 7.1: Página principal de la herramienta

1. Seleccionar la opción *"Node"* en el menú superior de la herramienta.
2. Elegir o escribir el tipo de nodo deseado, como se muestra en la imagen 7.2
3. Agregar los atributos correspondientes al nodo, tal como se muestra en la imagen 7.3
4. Una vez completado, el cliente es redirigido nuevamente a la página principal.
5. Ahora es posible visualizar el nodo creado en la página principal, tal como se muestra en la imagen 7.4

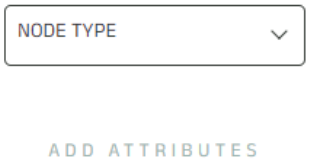

Figura 7.2: Elección de tipo de Nodo

Figura 7.3: Atributos de Nodo

Figura 7.4: Nodo creado visualizado en pantalla principal

Figura 7.5: Nodos tipo Persona sin relación

7.2. Creación de relación Nodo a Nodo

En este escenario, los dos nodos que se muestran la imagen 7.5 tendrán una relación de tipo amistad. A continuación se describen los pasos para llevar a cabo dicha relación:

1. En el menú superior de la herramienta, se selecciona la opción *Relate"*.
2. Se elige el nodo de origen y el nodo de destino para establecer la relación como se muestra en la imagen 7.6
3. Se selecciona el tipo de relación deseado entre los nodos, como se muestra en la imagen 7.6
4. Se eligen los atributos relacionados con la relación establecida como se muestra en la imagen 7.7
5. Finalmente, tanto el cliente como Neo4j permiten visualizar la relación creada como se muestran en las imágenes 7.8 y 7.9

Figura 7.6: Tipo y dirección de relación

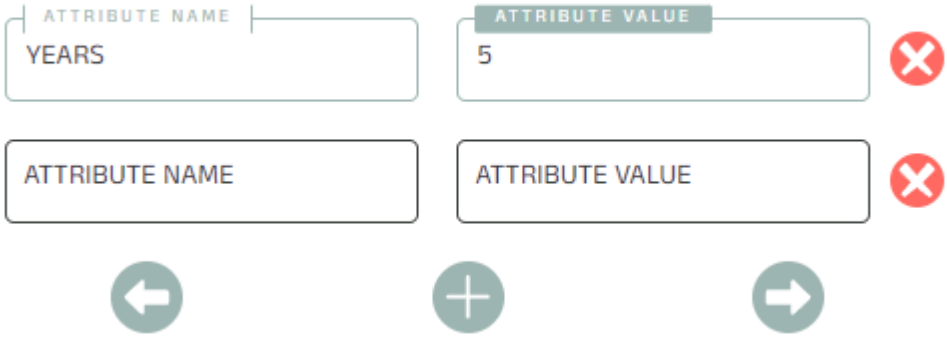

Figura 7.7: Atributos de la relación

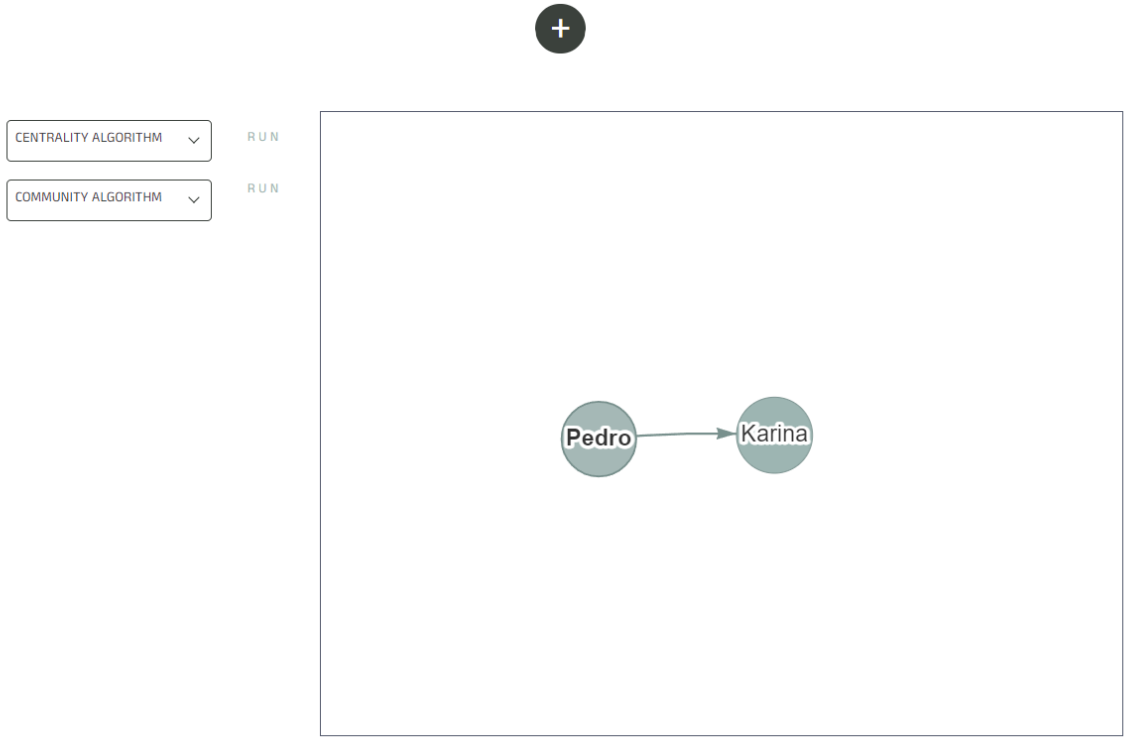

Figura 7.8: Relación visualizada de lado del cliente

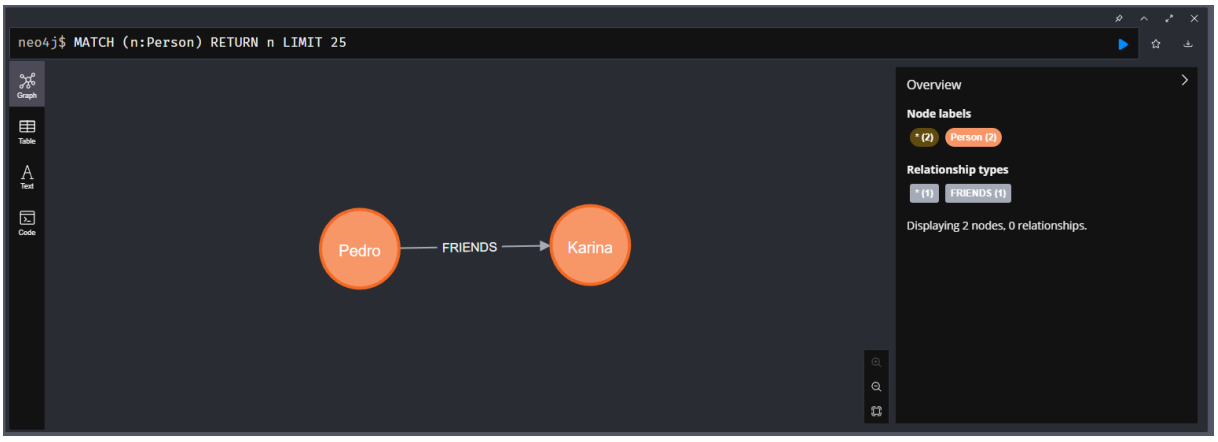

Figura 7.9: Relación visualizada de lado de Neo4J

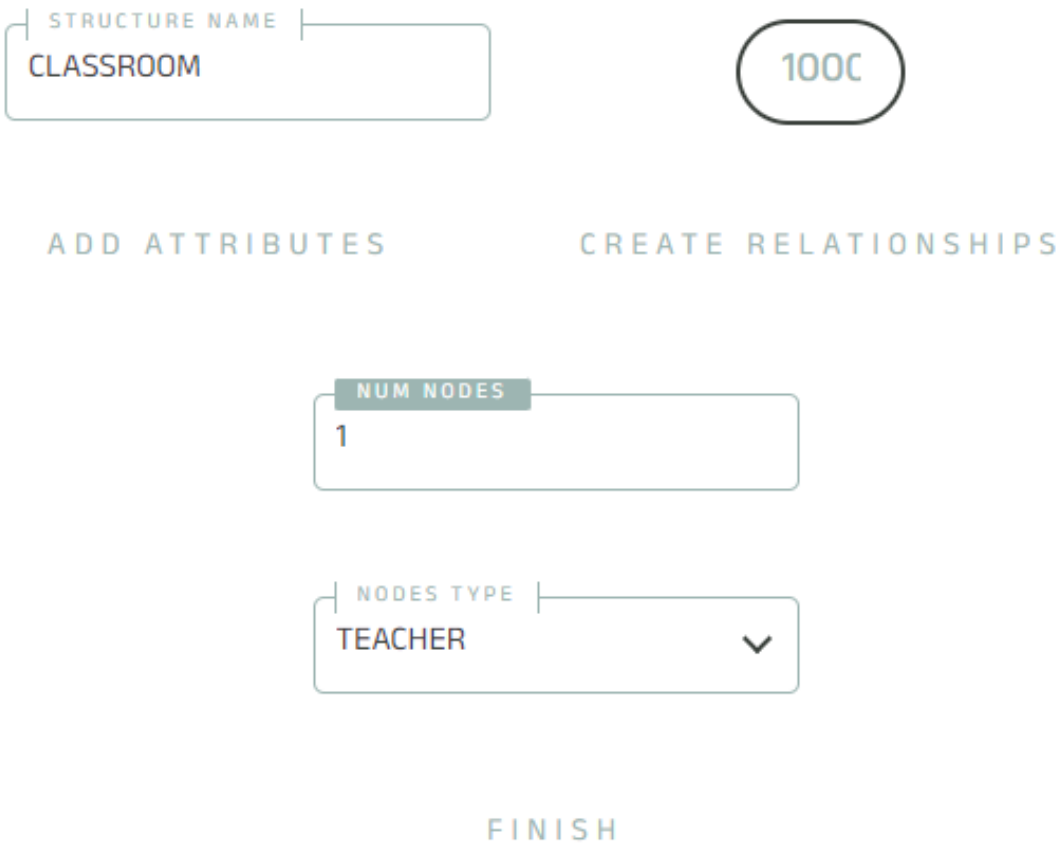

Figura 7.10: Página de creación de estructuras

7.3. Creación de 1000 salones de clase

En este escenario se presenta el proceso para crear 1000 salones de clases, donde cada salón tiene un maestro que enseña a 5 alumnos:

1. En el menú superior de l herramienta se selecciona la opción *"Manual"*.
2. Una vez redirigido a la pantalla de creación de estructuras, se ingresa el nombre de la estructura y la cantidad de estructuras que deseamos crear, como se muestra en la imagen 7.10
3. Se selecciona la opción de tener la cantidad de un solo nodo de tipo *"teacher"*, como se muestra en la imagen 7.10

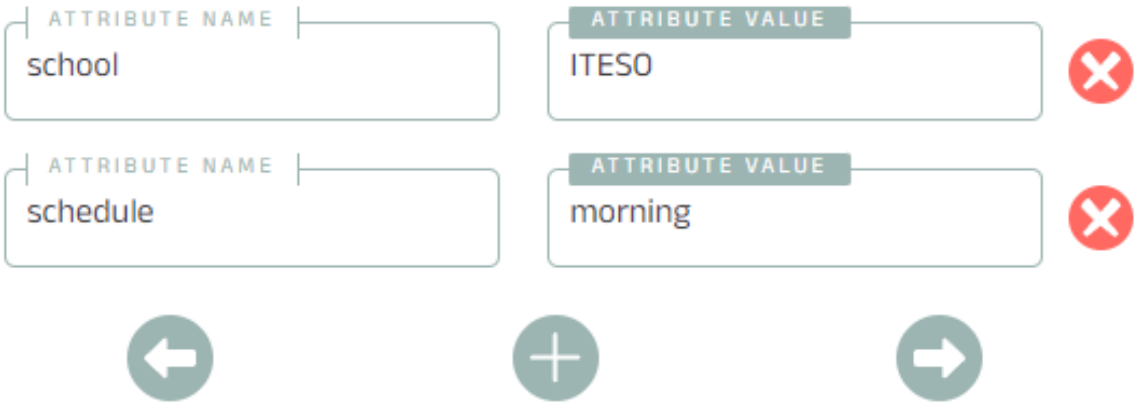

Figura 7.11: Atributos del nodo *TEACHER*

4. Damos click a la opción de *"Add attributes"* para agregar nuestros atributos al nodo, como se muestra en la imagen 7.11

5. Repetimos los dos pasos anteriores para crear 5 nodos de tipo *"student"*, como se muestra en las imágenes 7.12 y 7.13

6. Una vez redirigidos a la página de creación de estructuras procedemos a agregar nuestras relaciones dando click en *"Add relationships"* y consecuentemente en *Relate to All"*, donde indicaremos nuestro nodo *"Teacher"* como nodo origen y de tipo *"TEACHES"*, de esta manera estamos indicando al sistema que el nodo *"TEACHER"* tendra una relación a cada nodo *"STUDENT"* de tipo *"TEACHES"* como se muestra en la imágenes 7.14, 7.15 y 7.16

7. Automáticamente nos redirigiremos a la pantalla de creación de estructuras y damos click en *"Finish"*, esto mandará la petición a Neo4J para la creación de las estructuras y nos redirigirá a la página principal de la herramienta.

8. Una vez que regreses a la pantalla principal, se visualizarán algunas de las estructuras, como se muestra en la imagen 7.17 Sin embargo, si accedes a la base de datos de Neo4j, se puede observar en la imagen 7.18 que se han generado 6000 nodos y 5000 relaciones. Cada estructura cuenta con 1 nodo de tipo maestro relacionado con 5 nodos de tipo estudiante.

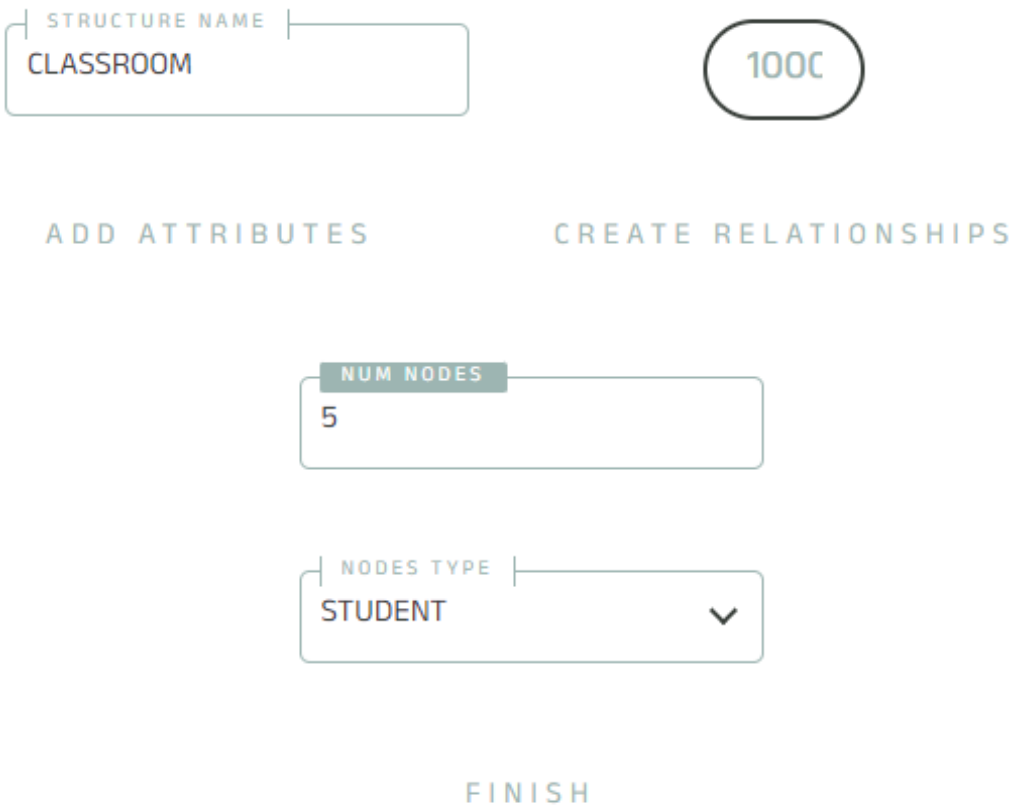

Figura 7.12: Creación de 5 nodos de tipo *STUDENT*

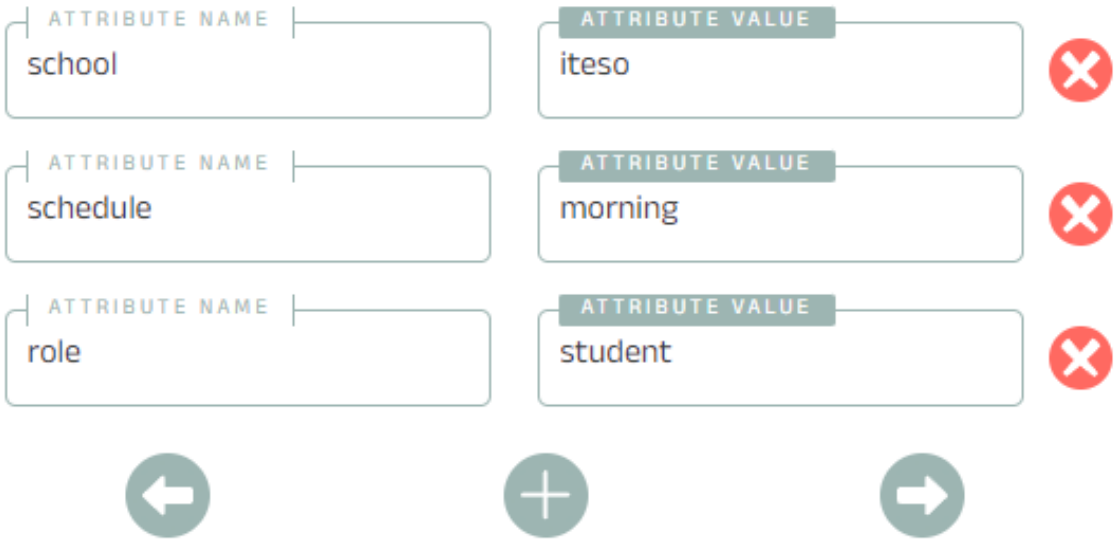

Figura 7.13: Atributos del nodo *STUDENT*

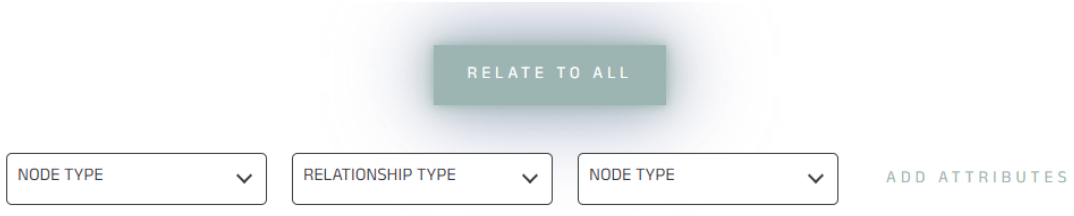

Figura 7.14: Selección de relación para estructuras

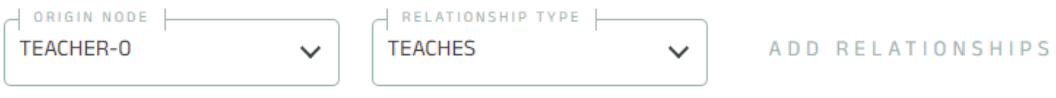

Figura 7.15: Relacionar todos los nodos

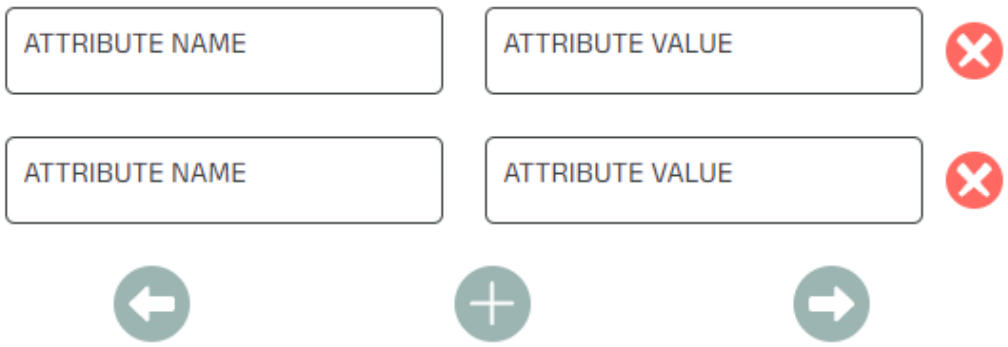

Figura 7.16: Atributos de la relación *TEACHES*

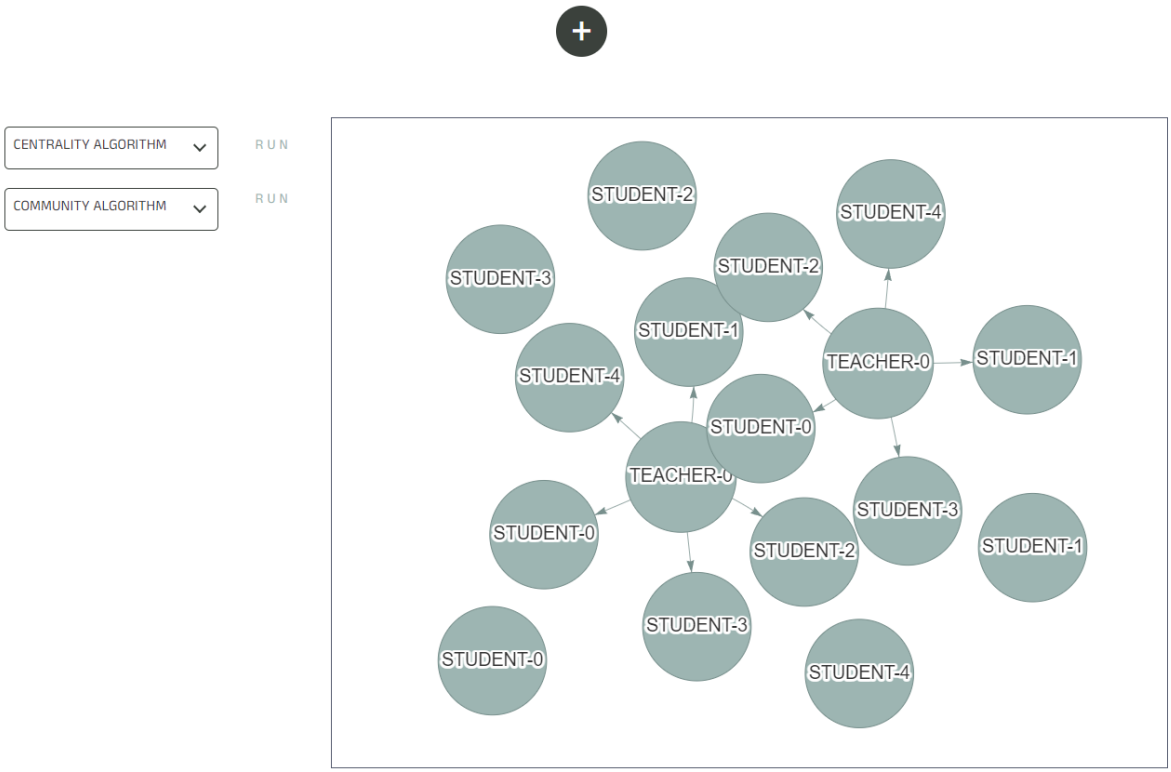

Figura 7.17: Página principal con estructuras creadas

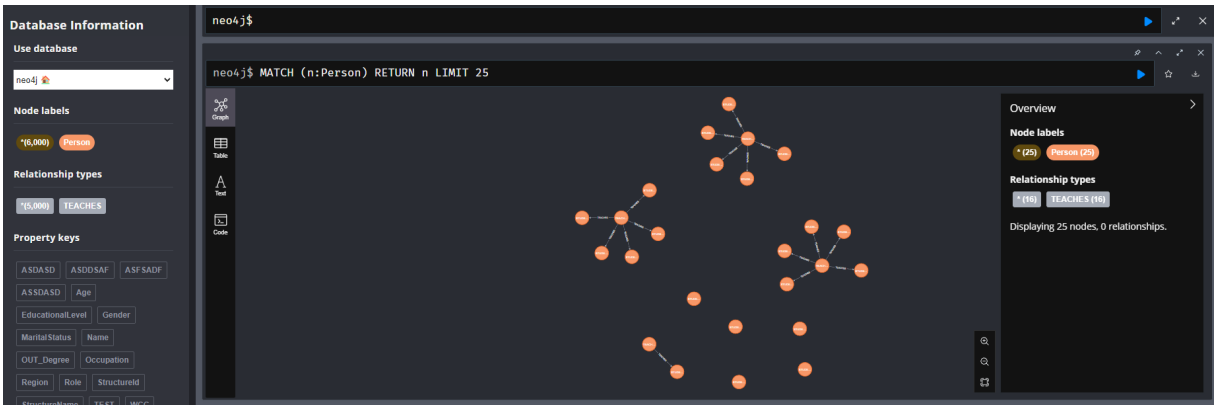

Figura 7.18: Estructuras generadas en Neo4J

BIBLIOGRAFÍA

[1] J. O. Gutierrez-Garcia y L.-F. Rodríguez, "Social determinants of police corruption: toward public policies for the prevention of police corruption," *Policy Stud*, vol. 37, n.º 3, págs. 216-235, 2016.

[2] M. A. Casar, "Anatomía de la Corrupción," vol. 3, pág. 115, 2020.

[3] H. Situngkir, "The Structural Dynamics of Corruption," pág. 13, 2014.

[4] D. Frąszczak, "RPaSDT—Rumor Propagation and Source Detection Toolkit," *SoftwareX*, vol. 17, pág. 100 988, 2022, ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2022.100988. dirección: https://www.sciencedirect.com/science/article/pii/S2352711022000085.

[5] A. Soto, H. Mora y J. A. Riascos, "Web Generator: An open-source software for synthetic web-based user interface dataset generation," *SoftwareX*, vol. 17, pág. 100 985, 2022, ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2022.100985. dirección: https://www.sciencedirect.com/science/article/pii/S2352711022000073.

[6] M.-n. JuliÃ¡n y T. Bonavia, "Psychological variables related to corruption: a systematic review," en, *Anales de PsicologÃa*, vol. 36, págs. 330-339, sep. de 2020, ISSN: 0212-9728. dirección: http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S0212-97282020000200016&nrm=iso.

[7] Oracle. "Definición de base de datos orientada a grafos." (2023), dirección: https://www.oracle.com/cl/autonomous-database/what-is-graph-database/ (visitado 01-07-2023).

[8] C. C. P. Alvarez, C. Pinilla y M. Bello, "Bases de datos orientadas a grafos," *Tecnología Investigación y Academia*, vol. 5, n.º 2, págs. 153-160, 2017.

[9] Neo4J. "Neo4J use cases." (2023), dirección: https://neo4j.com/use-cases/ (visitado 09-07-2023).

[10] Amazon. "Amazon Neptune." (2023), dirección: https://aws.amazon.com/neptune/ (visitado 09-07-2023).

[11] JanusGraph. "Janus Graph." (2023), dirección: https://janusgraph.org (visitado 09-07-2023).

[12] Neo4J. "Page Rank." (2023), dirección: https://neo4j.com/docs/graph-data-science/current/algorithms/page-rank/ (visitado 09-07-2023).

[13] Neo4J. "Degree Centraility." (2023), dirección: https://neo4j.com/docs/graph-data-science/current/algorithms/degree-centrality/ (visitado 09-07-2023).

[14] Neo4J. "Betweenness Centrality." (2023), dirección: https://neo4j.com/docs/graph-data-science/current/algorithms/betweenness-centrality/ (visitado 09-07-2023).

[15] Neo4J. "Louvain." (2023), dirección: https://neo4j.com/docs/graph-data-science/current/algorithms/louvain/ (visitado 09-07-2023).

[16] Neo4J. "Label Propagation." (2023), dirección: https://neo4j.com/docs/graph-data-science/current/algorithms/label-propagation/ (visitado 09-07-2023).

[17] Neo4J. "Weakly Connected Components." (2023), dirección: https://neo4j.com/docs/graph-data-science/current/algorithms/wcc/ (visitado 09-07-2023).