
**Universidad de las Ciencias Informáticas**

**"Facultad de Ciencias y Tecnologías Computacionales"**

# Título: Detección de comunidades en redes híbridas en el campo de la microfluídica

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

**Autores:** Roberto Primiano Rios
Adrián Figueroa Salvador

**Tutores:** Dr. Jorge Gulín González
Ing.Isabel Esther Rodríguez

## DEDICATORIA

A mi familia, por haber participado todos de una manera u otra en posibilitar mi desarrollo personal y aprendizaje en la universidad, sobre todo a mi madre, que ha luchado incansablemente para que pudiese lograr cada aspiración o sueño por más pequeño que fuera; a mi futura esposa Marla por todo el apoyo y levantarme el ánimo en cada noche de estrés, y además a todos mis amigos por estos años de cariño y momentos mágicos, tanto los buenos como los malos.

**Agradecimientos**

A mis tutores, en especial al doctor Gulín por toda la paciencia, dedicación y ayuda a lo largo de este proceso; a Neida, Kuroky, Nilda; sin su apoyo incondicional este trabajo no hubiese sido posible. Muchas gracias a todos.

**RESUMEN**

En el mundo real, múltiples tipos de objetos están interconectados, creando redes de información heterogéneas. Una red híbrida es una red heterogénea con una variedad de tipos de nodos y una variedad de tipos de relaciones. El concepto de red heterogénea enfatiza la complejidad a nivel estructural de la red y la riqueza de funciones estudiadas. Las características de la red híbrida de múltiples nodos y relaciones se reflejan principalmente en los dos aspectos siguientes: primero, la diversidad de nodos, incluida la variedad de tipos de nodos y, segundo, la riqueza de las relaciones entre ellos. La detección de estructuras comunitarias es un método que se utiliza para identificar grupos de nodos en una red. La detección de estructuras comunitarias es la característica estructural más estudiada de las redes complejas. Por otro lado, el abordaje de un nuevo tema investigativo en cualquier campo científico se torna complejo cuando no existe un conocimiento previo sobre las principales publicaciones o autores que han trabajado la temática. Por otra parte, a pesar de que existen aplicaciones informáticas para realizar estos estudios, en nuestro país aún no se ha desarrollado dicha solución y las accesibles internacionalmente son en la mayoría de los casos propietarias. Este trabajo se centra en la aplicación de algoritmos de detección de comunidades sobre una red híbrida de publicaciones en el campo de la microfluídica para corroborar que la metodología propuesta conduce a resultados positivos que contribuirán con las tareas de investigación científica. De los existentes algoritmos se decidió utilizar el algoritmo de propagación de etiquetas y el algoritmo de Louvain, ambos centrados en la estructura de la red. Los resultados preliminares obtenidos demuestran las potencialidades de los métodos de detección de comunidades para enfrentar problemas similares en ese o en otros campos científicos y académicos.

**Palabras clave:**
Microfluídica, Detección de comunidades, red híbrida, modularidad

**Abstract**

In the real world, multiple types of objects are interconnected, creating heterogeneous information networks. A hybrid network is a heterogeneous network with a variety of types of nodes and a variety of types of relationships. The concept of heterogeneous network emphasizes the complexity at the structural level of the network and the richness of studied functions. The characteristics of the hybrid network of multiple nodes and relationships are mainly reflected in the following two aspects first, the diversity of nodes, including the variety of node types, and second, the richness of the relationships between them. Community structure detection is a method used to identify groups of nodes in a network. The detection of community structures is the most studied structural feature of complex networks. On the other hand, the approach to a new research topic in any scientific field becomes complex when there is no previous knowledge of the main publications or authors who have worked on the subject. On the other hand, although there are computer applications to carry out these studies, in our country such a solution has not yet been developed and the ones available internationally are in most cases proprietary. This work focuses on the application of community detection algorithms on a hybrid network of publications in the field of microfluidics to corroborate that the proposed methodology leads to positive results that will contribute to scientific research tasks. From the existing algorithms, it was decided to use the label propagation algorithm and the Louvain algorithm, both focused on the network structure. The preliminary results obtained demonstrate the potential of community detection methods to face similar problems in this or other scientific and academic fields.

**Keywords:**

Microfluidics, Community detection, hybrids networks, modularity

TABLA DE CONTENIDOS

## Índice de ilustraciones:

## Índice de Ecuaciones

# INTRODUCCIÓN

El proceso evolutivo humano ha hecho que nuestra especie forme comunidades para adaptarse a su condición en el planeta. Esto llamó la atención de muchos estudiosos de diferentes ramas científicas y particularmente al estudio de patrones que existían en las relaciones entre los individuos que conformaban las distintas comunidades internas de las redes sociales, y fue tal el interés que despertó el Análisis de Redes Sociales (ARS) que los artículos con respecto al tema entre los años 1990 y 2005 alcanzaban las 3000 publicaciones (González 2019).

Jorge Dettmer González propone tres etapas para la evolución del ARS, en la primera etapa menciona el estudio llevado a cabo por autores que basaron sus trabajos de análisis utilizando la teoría de grafos creada por Leonhard Euler en el siglo XVIII y desarrollada con el paso de los años para optimizarla en la aplicación a problemas más complejos del que Euler trató.

Además de las aplicaciones mencionadas anteriormente algunas de las más recientes apuntan principalmente a la optimización de procesos o identificación de grupos con rasgos significativos comunes, como es el caso del estudio realizado en Chile que, tomando como muestra el claustro de los departamentos de Lenguaje y Matemáticas de tres escuelas, identifica los principales líderes en el proceso de la mejora educativa(Queupil, Montecinos 2020). Además, vale la pena mencionar un estudio realizado en Ecuador para notar la relación entre las complejidades no visibles en la baja producción científica de universidades, o sea se tomó todo tipo de información para identificar los factores que más perjudican la producción(Ponce Ordóñez et al. 2019); como último ejemplo sobre el panorama actual con respecto al ARS: la identificación de campos de investigación sobre el ARS combinando el análisis de co-citación y co-palabras, el cual significa una buena fuente de bibliografía con respecto al tema que nos aborda(Galvez 2018).

El estudio de redes de colaboración científica ha cobrado cada vez más importancia debido al aporte que trae a la comunidad científica en cuanto a la comprensión de los descubrimientos e innovaciones(Divakarmurthy, Menezes 2013), y se ha convertido en una herramienta importante de la cienciometría. El agrupamiento en comunidades es útil para detectar temas de investigación y revelar estructuras y dinámicas científicas, lo que significa una mejora sistemática de la comprensión de campos de investigación(Zhang et al. 2016; Wang et al. 2013).

Las redes sociales pueden ser representadas como grafos, tomando los individuos como nodos y las relaciones existentes entre ellos como las artistas (Wasserman, Faust 1994). Para el cual tomaremos el concepto de grafo brindado por (Ortega, Meza 1993). Un grafo G es una terna (V, E, φ), donde V es un conjunto finito de elementos llamados vértices o nodos del grafo, E es un

conjunto finito de elementos llamados lados y φ es una función que asigna a cada elemento de E un par de elementos de V.
Partiendo del concepto anterior y la proposición de Wasseman y Faust (Wasserman, Faust 1994) se puede analizar matemáticamente una red social como un grafo k-partito en que los vértices se dividen en k subconjuntos disjuntos, y cada borde conecta vértices en distintas particiones. Por tanto, un grafo bipartito es aquel que presenta dos conjuntos donde cada nodo de una de sus particiones tiene relación con un nodo de la otra.

Una red híbrida es una red heterogénea que contiene varios tipos de nodos o varios tipos de relaciones. Este concepto se centra en la complejidad a nivel estructural de la red, mientras que la red formada por múltiples nodos y múltiples relaciones enfatiza en la calidad de las funciones. Las características de las redes híbridas multi-nodo y multi-relaciones están reflejadas en dos aspectos principales: La diversidad de los nodos y la calidad de las relaciones donde los nodos pueden representar autores, literaturas, palabras claves y las relaciones pueden ser las citaciones o colaboraciones.

La detección de comunidades es un problema que abarca el ARS, utilizado para encontrar nodos en una red que cumplan con ciertos requisitos. Esta es la característica estructural más estudiada de las redes complejas. En la actualidad el estudio basado en la detección de comunidades se enfoca en dos objetivos principalmente: Extender los algoritmos existentes para aplicarlo a una red híbrida, y el otro enfoque está en disminuir las dimensiones de la red híbrida para transformarla en una red homogénea y luego realizar la detección de comunidades en esta nueva red(Parthasarathy, Shah, Zaman 2019).
Las aplicaciones de la detección de comunidades tienen cabida en todos los campos conocidos, ya que su base está en el reconocimiento de patrones, los cuales están presentes en todo lo que conocemos, por tanto, tiene utilidad en cualquier actividad. Por ejemplo en ciencias poco formalizadas como la Medicina, Sociología, Geociencias, Criminalística (Ruiz-Schucloper 2009).En Cuba la principal aplicación que utiliza la detección de comunidades es a partir de redes de autoría en grafos RDF (Ortiz Muñoz, Hidalgo Delgado 2016).

La microfluídica es la ciencia ingenieril donde el comportamiento de los fluidos dista de la teoría del flujo convencional debido a la pequeña escala del sistema creado utilizando conocimientos de distintas disciplinas científicas como la física, química, biología, ingeniería etc(Nguyen, Wereley, Shaegh 2019).
Las principales utilidades de la microfluídica se ven en el campo de la medicina y en la farmacología, como es su estudio para la obtención de nanomedicinas para tratar el cáncer, el análisis celular por la facilidad de analizar células sin riesgo de lisis, creación de sustancias médicas como hidrogel para heridas cutáneas (Sósol-Fernández et al. 2012), e incluso en la agricultura al analizar los nutrientes del suelo (CORDIS 2020).

Tras una investigación en distintas revistas cubanas de diferentes disciplinas científicas se encontraron aplicaciones en el campo de la Física, primeramente en una introducción al campo de la microfluídica (D. Fernández Rivas 2011) y por otra parte con los investigadores del Centro de Matemática Computacional de la Universidad de Ciencias Informáticas como principales autores a cargo de proyectos nacionales, internacionales e institucionales, trabajos como “Caracterización computacional de la difusión en un sistema unidimensional de partículas interactuantes en régimen hidrodinámico” (Gulín-González et al. 2014), “Hydrodynamical characterization of red blood cells interactions in a high confinement regime. A computational study” (J. Gulín-González, E. Navas-Conyedo 2017), “Herramienta de visualización dinámica de simulaciones del proceso de difusión en microfluidos con componentes biológicos”. En la prensa periódica no especializada en ciencias, el periódico Granma se refirió a la creación de hidrogeles (Granma 2022).

El principal problema de los estudios de este campo es que su desarrollo puede ser mediante experimentos o teórico-computacionalmente, y en ambas formas es costoso porque involucra un gran número de partículas y múltiples interacciones. Además, al ser un campo científico relativamente nuevo hay mucho camino que recorrer en cuanto a estudios sobre la microfluídica como campo de acción, por ello, un ARS tomando las publicaciones científicas como objeto de estudio puede llegar a identificar comunidades de colaboración académicas mediante redes híbridas de investigadores, palabras clave, revistas, colaboraciones y citas.
Actualmente existe una amplia red de documentos científicos entre los que se incluyen los enfocados en el campo de la microfluídica, la búsqueda de información sobre la microfluídica en estas redes se hace tediosa debido a la gran cantidad de publicaciones, las cuales no están ni clasificadas ni organizadas, por lo que se desperdicia tiempo de investigación solamente en la obtención de material de estudio volviendo aún más costosas las investigaciones científicas además que existe el riesgo de pérdida de información o una gestión ineficiente de la misma.

Teniendo en cuenta lo expuesto anteriormente, se plantea como **problema científico**: ¿Cómo detectar comunidades de colaboración académicas en el campo de la micofluídica con base en redes híbridas?
Se propone como **objetivo general**: Desarrollar un sistema informático para la detección de comunidades en el campo de la microfluídica a partir de la construcción de redes híbridas y de la aplicación de los modelos computacionales y algoritmos existentes en la literatura.

Del **objetivo general** se derivan los siguientes objetivos específicos:

1. Sistematizar los referentes teóricos y metodológicos asociados con las investigaciones en el campo de la microfluídica y los algoritmos de detección de comunidades en redes híbridas.
2. Construir una red híbrida de investigadores, revistas y relaciones de colaboración entre autores y cita de autores, en el campo de la microfluídica.
3. Aplicar los algoritmos de detección de comunidades a las redes híbridas construidas.
4. Analizar los resultados de la aplicación de los algoritmos de detección de comunidades en redes híbridas.
5. Validar la solución propuesta a partir de los métodos definidos en la investigación.

Para dar respuesta a las preguntas científicas, guiar la investigación y garantizar la solución al problema expuesto con anterioridad, se planifican las siguientes **tareas científicas**:

1. Elaboración de un marco teórico referencial con los conceptos fundamentales que caracterizan objeto de estudio de la investigación: Los modelos computacionales y algoritmos de detección de comunidades.
2. Identificación y caracterización de sistemas similares existentes en el mundo para la confección de la solución propuesta.
3. Selección y caracterización de las herramientas y tecnologías a utilizar en el proceso de detección de comunidades en redes híbridas.
4. Implementación de las funcionalidades definidas para el sistema.
5. Realización de pruebas al sistema para la verificación del cumplimiento de las funcionalidades en la aplicación implementada.

El **objeto de estudio** de la investigación son los modelos computacionales y algoritmos para detección de comunidades en redes híbridas de investigadores, palabras clave y revistas y relaciones de colaboración entre autores y cita de autores.

El **campo de acción** queda enmarcado en los algoritmos utilizados en la detección de comunidades en redes híbridas.

Se parte de la **hipótesis** de que si se logran aplicar algoritmos de detección de comunidades a redes híbridas en el campo de la microfluídica será posible obtener información relevante sobre los grupos de investigadores, temas de investigación y relaciones de colaboración en este campo.

**Métodos propuestos para desarrollar la investigación**

Se propone la utilización de los siguientes métodos de investigación: empíricos, teóricos y estadísticos.

**El método empírico de observación científica:** para determinar el estado inicial del objeto de estudio, la identificación de necesidades en el estudio de las redes de investigadores en el campo de la microfluídica.

**El método teórico de sistematización:** Se utilizará en el ordenamiento y clasificación de la información sobre los modelos computacionales y algoritmos para detección de comunidades en redes híbridas de investigadores, palabras clave y revistas y relaciones de colaboración entre autores y cita de autores.

**El método teórico hipotético-deductivo**: En la definición de la hipótesis sobre el desarrollo del(los) algoritmo(s) a utilizar para el estudio de detección de comunidades.

El **método estadístico** de análisis de datos: Para el análisis y sistematización de los datos de las investigaciones en el campo de la microfluídica y el análisis de las redes híbridas.

**Aportes de la investigación**

**Aporte Teórico**:

La aplicación de los diferentes algoritmos para la detección de comunidades en el campo de la microfluídica computacional y su comparación.

**Aporte Práctico**:

El estudio permitirá el análisis de los principales temas, grupos, investigadores y redes de investigadores en el campo de la microfluídica, particularmente la computacional. En este sentido el desarrollo de la tesis facilitará la toma de decisiones con relación a futuros temas de investigación en este campo.

A partir de lo planteado anteriormente se llegó a la conclusión que utilizando la detección de comunidades en redes híbridas de publicaciones científicas enfocado al campo de la microfluídica será provechoso para la clasificación tanto de publicaciones como de grupos de científicos dispuestos en dicho campo, permitirá una mejor organización de la información y una gestión más

eficiente de la misma, disminuirá el tiempo empleado en la búsqueda de dicha información, además de aportar posibles temas de investigación que están relacionados con la microfluídica. Por lo tanto, es una aplicación valiosa para Cuba, ya que lo propuesto anteriormente, proveerá a los científicos cubanos una herramienta para encontrar colegas de investigación y trabajo en sus proyectos.

**Novedad:**

Se presenta por primera vez un estudio sobre detección de comunidades en el campo de la microfluídica, basado en diferentes algoritmos disponibles en la literatura.

# CAPÍTULO 1: FUNDAMENTOS DE LA INVESTIGACIÓN.

## Introducción:

En este capítulo se presentarán los principales fundamentos teóricos relacionados con la detección de comunidades, así como una breve reseña de los métodos de detección de comunidades existentes y el panorama actual de este campo científico.

### Conceptos relacionados:

**Comunidad:**

El término de comunidad puede ser confuso si buscamos, ya que es un tema que ha llamado la atención de muchos estudiosos en distintos campos, y le han asignado un significado dependiendo de su área de estudio, o sea hay casi tantos conceptos de comunidad como disciplinas científicas. Por lo que ciñéndonos a un concepto más general una comunidad es un conjunto de individuos que comparten rasgos.(Warz, Verdugo sin fecha).

En la temática que nos aborda la comunidad es tratada como un conjunto de nodos de una red que están densamente conectados entre sí que con el resto de la red(Berrocal, Figuerola, Medrano sin fecha) y a esta red general se le llama red social.

**Detección de Comunidades:**

La detección de comunidades se refiere a la técnica de identificar comunidades o particiones en un grafo que representa una red social en dependencia de las relaciones entre dichos nodos.(Ortiz Muñoz, Hidalgo Delgado 2016).

**Red Híbrida:**

Una red híbrida es una red heterogénea que contiene varios tipos de nodos o varios tipos de relaciones. Este concepto se centra en la complejidad a nivel estructural de la red, mientras que la red formada por múltiples nodos y múltiples relaciones enfatiza en la calidad de las funciones. Las características de las redes híbridas multi-nodo y multi-relaciones están reflejadas en dos aspectos principales: La diversidad de los nodos y la calidad de las relaciones donde los nodos pueden representar autores, literaturas, palabras claves y las relaciones pueden ser las citaciones o colaboraciones.

**Red Social:**

Una red social es un conjunto de personas con algunos patrones de interacción entre ellos. Algunas de las redes sociales estudiadas en la historia son los patrones de amistad entre individuos, relaciones comerciales entre compañías, matrimonios entre familias.(M. E. J. Newman Reviewed 2003).

Matemáticamente hablando, una red social es representada con un grafo. Uno de los más comunes es el grafo k-partito, donde los vértices o nodos son particionados en k-conjuntos

disjuntos, donde todos los nodos de uno de estas particiones tiene una relación con al menos un individuo de otra partición.(Jiang et al. 2021).

# 1 DETECCIÓN DE COMUNIDADES

## 1.1 Técnicas tradicionales de detección de comunidades:

- **Partición de grafos:** Consiste en dividir los vértices en g grupos de tamaño predefinido, de tal manera que los vértices se dividan en grupos de tamaño predefinido., de modo que el número de enlaces en un bloque es más denso que el número de aristas entre grupos. (Fortunato 2010).
- **Agrupación jerárquica:** Basada en la medida de similitud de vértices. No necesitan un tamaño y número predeterminado de comunidades. Se pueden representar mejor con dendogramas.(Papadopoulos, Kompatsiaris, Vakali 2010)
- **Algoritmos divisorios:** Consiste en eliminar las aristas entre los grupos de la red sobre la base de la baja similitud para separar las comunidades entre sí. (Murata, Ikeya 2010)
- **Agrupamiento parcial:** Dividir un conjunto de datos en un número predeterminado de n agrupaciones no solapadas. El objetivo de esta técnica es dividir los puntos de datos en n agrupaciones para mejorar la función de costo basada en la medida de diferencia entre nodos.(Jolliffe 2002)
- **Agrupamiento espectral:** Cubre todas las técnicas que usan valores propios para segmentar un conjunto de puntos de datos en función de su similitud por pares.(Fortunato 2010; Dhumal, Kamde 2015).

## 1.2 Técnicas de detección de comunidades basadas en la optimización de la modularidad:

Las técnicas basadas en la optimización de la modularidad, tienen como base agrupar de distintas formas los nodos y comprobando la variación de la modularidad. Aun siendo la base de todas ellas, el enfoque varía en dependencia del algoritmo, por ejemplo:

- **Técnicas codiciosas:** Esta es una técnica de fusión, donde cada nodo pertenece inicialmente a un módulo diferente y luego se fusiona iterativamente de acuerdo con la ganancia del parámetro.
- **Recocido simulado:** Es un método estocástico separado que se utiliza para la optimización global de la función objetivo especificada (Guimerà, Marta, Amaral 2007).
- **Optimización extrema:** Esta técnica se enfoca en optimizar las variables locales. Se ha utilizado en (Duch, Arenas 2005) para la optimización de la modularidad.

- **Optimización espectral:** Se refiere al uso de vectores propios y valores propios de la matriz de modularidad para optimización de la modularidad (Fortunato 2010).

### 1.2.1 Técnicas de detección de comunidades solapadas

Uno de los principales problemas de la detección de comunidades consiste en la existencia de solapamiento en el sistema. Actualmente, la técnica más utilizada para la identificación de comunidades solapadas es la percolación de camarillas. La base de esta técnica es que es más probable que se formen las camarillas a partir de aristas internas densamente conectados que de aristas exteriores débilmente conectados. Las comunidades se conforman por k-cliques que indican el conjunto de subgrafos de k-vértices. Las comunidades consisten en k-cliques (muestra subgráficos completos de k-vértices). La comunidad k-clique es un componente enorme que consta de todas las k-clique adyacentes conectadas como una serie k-clique (Macropol, Singh 2010).

### 1.2.2 Algoritmos de detección dinámica de comunidades

**Modelo Potts:** Está basado en una generalización del modelo de Ising de la física estadística. En este caso, las variables de rotación de Potts se pueden asignar a los nodos en el grafo con estructura de comunidad. Mediante la interacción de ciclos vecinos, se puede determinar la estructura de la comunidad a partir de grupos de similar valor en el sistema, ya que habrá más interacciones en la comunidad y menos interacciones fuera de ella.

**Paseo Aleatorio:** Un caminante comienza a moverse en una comunidad de un nodo y cada vez que camina a un nodo adyacente se elige aleatoria y uniformemente.(Hughes 1996)

**Sincronización:** Se trata de un fenómeno emergente que ha despertado el interés de diversos campos. Se produce en unidades que interactúan entre sí y son persuasivas en naturaleza, tecnología y sociedad. En el caso de sincronización, las unidades del sistema permanecen en su estado antiguo o similar a lo largo del tiempo. La sincronización también se utiliza para descubrir comunidades en la red.

En redes complejas a gran escala, los algoritmos de detección de comunidades dirigido por líderes (LDCD) son una nueva dirección en el diseño de algoritmos. La idea principal es identificar algunos nodos específicos en la red de destino, llamados nodos líderes, alrededor de los cuales se pueden calcular las comunidades locales. Al estar basados en cuentas locales, son especialmente interesantes para gestionar redes de gran tamaño.(Yakoubi, Kanawati 2014)

### 1.2.3 Métodos de detección de comunidades en redes híbridas de múltiples nodos y relaciones

La investigación sobre los métodos de detección de comunidades en redes híbridas de múltiples nodos y relaciones, actualmente se centra principalmente en las siguientes dos categorías: una es extender los algoritmos existentes para tratar directamente con redes híbridas, y la otra es reducir el tamaño de la red híbrida a una red homogénea y luego implementar la detección de comunidades (Berlingerio, Coscia, Giannotti 2011).

Con base en las dos ideas anteriores, los métodos de detección de comunidades en redes híbridas de múltiples nodos y reacciones se dividen principalmente en las siguientes cinco categorías: métodos basados en modelos probabilísticos, métodos basados en meta-rutas, métodos basados en nodos semilla, métodos de modularidad extendida y métodos homogéneos de redes heterogéneas.

### 1.2.4 Métodos basados en modelos probabilísticos

En el método basado en modelos probabilísticos, algunos algoritmos combinan el problema de clasificación y el problema de detección de comunidades y se complementan entre sí. El algoritmo RankClus (Sun et al. 2009) es el primer algoritmo propuesto basado en redes hibridas, solo es aplicable a dos tipos de nodos. (Sun, Han, Yu 2009) propusieron NetClus, un nuevo algoritmo basado en el algoritmo RankClus, que usa los enlaces entre múltiples nodos para generar agrupamientos de red de alta calidad. Dicho algoritmo tiene mejores efectos de agrupación, aunque es aplicable solo a la estructura de la red en estrella, y los objetos representativos del conjunto de datos deben conocerse de antemano. De esta forma, se ha mejorado el algoritmo RankClass (Ji, Han, Danilevsky 2011) para que sea adecuado a cualquier red híbrida, y puede hacer uso completo de la información de la etiqueta de cualquier objeto de datos.

Para revelar el proceso de evolución de cada tipo de nodo, (Gupta et al. 2011) propone el algoritmo EnetClus, el cual agrupa de manera evolutiva y usa un método de suavización del tiempo para mostrar los agrupamientos que cambian con el tiempo. El algoritmo OcdRank propuesto por (Qiu et al. 2015), tiene una baja complejidad temporal y admite la actualización incremental de datos.

Dado que el método basado en la clasificación debe predeterminar el número de comunidades, no es estable. Para hacer esto, se propuso un método de agrupación espectral de red híbrida para modelos de bloques aleatorios y se aplicó el algoritmo de maximización de la expectativa variacional para obtener una inferencia posteriormente, beneficioso para redes grandes, lo que permite que diferentes tipos de nodos tengan múltiples relaciones de pertenencia, pero este algoritmo no resuelve el problema de las comunidades superpuestas.

Por tanto, se puede observar que los métodos basados en modelos probabilísticos incluyen dos métodos: el basado en clasificación y el estadístico-probabilístico. Aunque los algoritmos basados en modelos probabilísticos están desarrollados de manera integral, son difíciles de aplicar a redes de gran escala debido a su complejidad en términos de espacio y tiempo. Además, la mayoría de las comunidades deben mapearse en función del conocimiento previo. Cuando la red es grande, es difícil predecir con precisión. La eficiencia de la detección de comunidades depende del número previamente estimado de comunidades, lo que conduce a la inestabilidad de los resultados.

### 1.2.5 Métodos basados en nodos semilla

Entre los métodos de detección de comunidades, se han convertido en tendencia los métodos basados en nodos semilla (Hmimida, Kanawati 2015a). La idea básica del método es identificar algunos nodos específicos en la red, que se denominan nodos semilla, y luego construir una comunidad alrededor de estos nodos (Kanawati 2011; Papadopoulos, Kompatsiaris, Vakali 2010)

El primer algoritmo de detección de comunidades basado en nodos semilla fue propuesto por Yakoubi. El objetivo es seleccionar los nodos más céntricos directamente como nodos semilla, realizar el cálculo de la comunidad local alrededor de dichos nodos y, seguidamente, llevar a cabo la detección de comunidad a partir de conjuntos de comunidades locales (Yakoubi, Kanawati 2014). Aunque este algoritmo solo es aplicable a redes homogéneas. (Hmimida, Kanawati 2015a) ampliaron el algoritmo Licod a las redes hibridas (fue nombrado mux-Licod). Este método tiene en cuenta los diferentes tipos de relaciones entre los nodos en diferentes capas de la red híbrida y los resultados experimentales muestran que tiene una buena rentabilidad.

El método basado en nodos semilla es de cálculo parcial, fácil de entender y adecuado para el procesamiento de redes a gran escala y redes dinámicas (Yakoubi, Kanawati 2014). No obstante, aún no existe un acuerdo sobre la manera de escoger los nodos semillas de manera eficiente. Asimismo, al fusionar comunidades sin nodos semilla, pueden existir problemas de agrupación excesiva de comunidades grandes y número excesivo de comunidades pequeñas.

### 1.2.6 Métodos basados en meta-ruta

Los nodos se encuentran conectados por múltiples enlaces. Dichos enlaces vinculan diferentes nodos con semántica diferente formando meta-rutas. Una meta-ruta es una serie de tipos de objetos que representan una relación semántica entre dos nodos (Li, Sun, Mao 2018), y es una herramienta eficaz de captura semántica que puede capturar información rica en redes híbridas (Sun et al. 2011; Shi et al. 2017).

PathSim es el primer algoritmo basado en meta-ruta, propuesto por (Shi et al. 2017). Fue propuesto para redes homogéneas y tiene un buen desenvolvimiento en la medición de la

semejanza entre nodos del mismo tipo. La mayoría de los métodos de detección de comunidades basados en meta-rutas tienen dos problemas fundamentales: La similitud obtenida directamente de la meta-ruta suele ser una desviación y el de cómo combinar la similitud de diferentes meta-rutas (Li, Sun, Mao 2018). Por tanto, basándose en la normalización de PathSim para eliminar las desviaciones de similitud, fue diseñado un componente de fusión flexible para optimizar de manera dinámica los resultados, de forma tal que los resultados de la detección de comunidades sean superiores. Un algoritmo de semejanza (HeteSim) fue propuesto por (Shi et al. 2012), el cual puede medir los mismos o diferentes tipos de nodos basados en meta-rutas. Dicho algoritmo calcula la semejanza a través de un recorrido aleatorio bidireccional y se emplea mejor en tareas de consulta y agrupación que los algoritmos tradicionales; no obstante, HeteSim solo se puede aplicar a un solo entorno. Debido a su campo de meta-ruta único, no puede capturar mucha información semántica en una red de información heterogénea y, debido a su alta complejidad, no es adecuado para redes a gran escala. Luego, (Meng et al. 2014) propusieron un procedimiento de doble recorrido aleatorio basado en una meta-ruta dada y una meta-ruta inversa para calcular la similitud entre dos objetos (AvgSim), el que se puede aplicar en una red a gran escala y tiene una mejor eficacia de agrupación.

Meta-rutas distintas contienen información diferente, y elegir diferentes meta-rutas acarreará a diferentes resultados para la detección de comunidades. Determinar la cantidad de meta-rutas específica o las meta-rutas óptimas entre varias meta-rutas es un problema difícil. (Sun et al. 2013) propusieron el algoritmo Pathselclus, que puede otorgar distintos pesos a diferentes meta-rutas en redes híbridas. (Shi et al. 2017) implantaron el método de paseo aleatorio HRank, basado en las meta-rutas para evaluar la importancia de los nodos y las meta-rutas, y los resultados experimentales muestran las ventajas únicas de las meta-rutas.

Los métodos de detección de comunidades basados en meta-ruta en redes híbridas han sido mejorados principalmente por el método PathSim para redes homogéneas. El método basado en meta-rutas es relativamente simple y directo, aunque la semejanza resultante suele ser una medida de desviación (Li, Sun, Mao 2018) . Asimismo, distintas meta-rutas contienen diferente información, por lo que se dificulta calcular correctamente la semejanza entre los nodos para mostrar una correcta correlación semántica y como elegir la meta-ruta recomendable entre múltiples meta-rutas para obtener el efecto de partición óptimo.

### 1.2.7 Métodos de modularidad extendida

La modularidad se introdujo por primera vez para valorar los resultados de la detección de comunidades. Con el desarrollo de nuevas investigaciones, han surgido algoritmos modulares de detección de comunidades (Newman, Girvan 2004; Tang, Wang, Liu 2009, p. 200; Nicosia et al. 2009). Se propuso por primera vez el algoritmo de optimización de la modularidad FN, por

(Newman, Girvan 2004), que considera cada nodo como una comunidad y calcula el valor de la modularidad después de combinar dos comunidades. Los métodos de correlación de la comunidad se aplican para amplificar el coeficiente máximo o reducir el coeficiente mínimo, y el descubrimiento de la comunidad se completa por iteración hasta que el coeficiente deja de aumentar (Newman, Girvan 2004).

(Guimerà, Marta, Amaral 2007) plantearon un algoritmo de modularidad extendido adecuado para redes bipartitas, que puede, de manera independiente, identificar nodos con conexiones de salida similares y nodos con conexiones semejantes. En (Murata, Ikeya 2010) se presenta un algoritmo de modularidad apropiado para redes k-core. Al igual que otros algoritmos de modularidad generales, este algoritmo tiene el problema de la limitación de resolución y no es adecuado para redes híbridas con morfología general. (Liu et al. 2014) propone un método de modularidad compuesta. La idea principal de este es descomponer una red heterogénea en múltiples subredes, integrar la modularidad en cada subred y optimizar la modularidad compuesta basada en el algoritmo de Louvain para lograr la detección de comunidades. Este algoritmo es conveniente para redes a gran escala y redes de morfología general, además de que no demanda conocimientos anteriores.

Los métodos de modularidad extendida evolucionaron a partir del algoritmo en una red homogénea y tienen una alta estabilidad. Sin embargo, este método tiene una alta complejidad temporal e inevitablemente limita la resolución máxima de la unidad, lo que hace imposible detectar pequeñas comunidades en redes de gran escala (Fortunato 2010; Fortunato, Barthélemy 2007).

### 1.2.8 Métodos homogéneos en redes heterogéneas

Dado que el método de detección de comunidades de redes homogéneas está bien establecido, la dimensión de las redes heterogéneas puede reducirse a redes homogéneas para utilizarse el método de detección de comunidades de redes homogéneas. Para lograr reducir la dimensión de las redes heterogéneas se utilizan principalmente métodos como la factorización de la matriz no negativa (Lee, Seung 1999), el modelo de temas (Blei 2003), análisis de componentes principales (Jolliffe 2002), el análisis discriminante lineal (Mika et al. 1999), entre otros.

### 1.2.9 Método de factorización de matrices no negativas

Los métodos de descomposición de matrices no negativas pueden descomponer una matriz no negativa dada en dos matrices no negativas (Liu, Gong, Tao 2016), que son la matriz base y la matriz de coeficiente, respectivamente La matriz de módulo se utiliza para reemplazar la matriz original para reducir el número de dimensiones. (Meng, Tafavogh, Kennedy 2014) propuso un

método de detección de comunidades de redes heterogéneas en base a la descomposición de matrices y las rutas semánticas. Zhang propuso un método de análisis de matriz de tres factores no negativo, HMFlus, que calcula la similitud e integra información entre organismo de la misma especie en HMFlus. Este método puede juntar a la vez todos los objetos en una red híbrida (Zhang et al. 2016). Un algoritmo de factorización alterna penalizada PAF, fue propuesto por Liu (2020), para resolver los problemas de optimización correspondientes desde la perspectiva de la descomposición de la matriz para la red de atributos multicapa. Este algoritmo no solo resulta efectivo en la detección de comunidades, también es aplicable a la morfología de la red.

### 1.2.10 Métodos del modelo temático

La introducción del modelo objetivo puede extraer la información objetiva oculta en la información del texto, para mejorar la eficiencia de descubrimiento de la comunidad.(Mei et al, 2018) fusionaron el modelo temático con el análisis de redes sociales, aprovecharon las ventajas del modelo temático estadístico y la regulación discreta al máximo, optimaron el modelo temático a través de la regularización y ejecutaron la detección de comunidades.

### 1.2.11 Métodos de análisis de componentes principales y de análisis discriminante lineal

El análisis de componentes principales es un método que tiene dos objetivos fundamentales: Encontrar la variabilidad de los datos y reducir las dimensiones de un conjunto de datos, siendo este último el que interesa en esta investigación. Para reducir las dimensiones de un conjunto de datos este método se centra en eliminar la redundancia e irrelevancia de la información analizando las variables recogidas, y llevándolas a un espacio completamente separado donde las variables iniciales se transforman en un conjunto de variables artificiales conocidas como componentes principales que carecen de relación entre ellas.

El análisis discriminante lineal es un método de clasificación supervisado de variables cualitativas en donde hay que conocer los grupos a priori, se centra en analizar y encontrar las diferencias entre objetos para caracterizar las clases, específicamente los puntos que separan una clase en particular de otra. Es simple y robusto además de arrojar resultados tan buenos como algoritmos más complejos. El objetivo es encontrar una regla que permite ordenar los objetos en **clases predefinidas** y construir un modelo que ayuda al usuario a descubrir patrones y ordenar los datos. Ambos métodos se utilizan para reducción lineal de la dimensión y utilizan métodos de proyección lineal para mapear los datos de alta dimensión a un espacio de baja dimensión. La diferencia entre ellos radica en que el primer método asegura que los datos después de reducir el tamaño retengan más información original, mientras que el segundo método hace que sea más fácil distinguir los datos después de la reducción de tamaño. Los estudios actuales aplican estos dos

métodos solo en redes de un solo nodo (Li et al. 2016; Yuan, Wang, Song 2016; Li et al. 2016) o en redes bipartitas(Liu, Chen 2013).

A pesar de que entender los métodos homogéneos en redes heterogéneas resulta fácil de entender, reducir las redes heterogéneas para tornarlas redes homogéneas y trabajar con la información puede complicarse, y como consecuencia provocaría una distorsión de los datos, además de que la implementación de estos métodos es bastante alta. Ya que una de las condiciones de algunos métodos es conocer de antemano la cantidad de clústeres, en comunidades a grande escala podrían no satisfacer las necesidades de la cuestión.

## 1.3 Indicadores de evaluación de efecto de detección de comunidades utilizados habitualmente

Hay varios tipos de indicadores y se utilizan en dependencia del método de detección de comunidades y necesidades experimentales. Los más utilizados son: La información Mutua Normalizada (NMI) (Lancichinetti, Fortunato, Kertész 2009), el índice de Rand Ajustado (ARI) (Santos, Embrechts 2009) y la Modularidad (Newman, Girvan 2004), los dos primeros indicadores se utilizan para resultados de detección de comunidades reales conocidas y la modularidad para las desconocidas.

La NMI es un método basado en la teoría de la información y la probabilidad para evaluar los resultados de una detección de comunidades. Para mostrar la calidad de los resultados detecta las diferencias entre los resultados reales de la detección.

$$NMI = \frac{-2\sum_{i=1}^{C_a}\sum_{j=1}^{C_b} N_{ij} * \log\left(\frac{N_{ij} * N}{N_i * N_j}\right)}{\sum_{i=1}^{C_a} N_i * \log\left(\frac{N_i}{N}\right) + \sum_{j=1}^{C_b} N_j * \log\left(\frac{N_j}{N}\right)}$$

*Ecuación 1 Información Mutua Normalizada*

En la ecuación 1: A y B son los conjuntos de resultados divididos de la red, N es el número total de nodos, Ca y Cb representan el número de comunidades, Ci(Cj) es la suma de los elementos de la fila i(j) en C. El rango de los valores está entre 0 y 1 y cuanto mayor sea el valor, más preciso será el resultado.

ARI mide la semejanza entre dos nodos analizando sin son consistentes bajo diferentes detecciones de comunidades:

$$ARI = a_{11}\frac{\frac{(a_{11}+a_{01})(a_{11}+a_{10})}{a_{00}}}{\frac{(a_{11}+a_{01})+(a_{11}+a_{10})}{2} - \frac{(a_{11}+a_{01})(a_{11}+a_{10})}{a_{00}}}$$

*Ecuación 2 Índice Rand Ajustado*

Donde a11 representa el número de pares de puntos que pertenecen a la misma comunidad en la comunidad real y la comunidad real, a00 representa el número de pares que no pertenecen a la misma comunidad en la comunidad real y la comunidad real, a10 representa el número de pares de puntos que pertenecen a la misma comunidad en la comunidad real pero no pertenecen a la misma en la comunidad real, a01 representa el número de pares de puntos que no pertenecen a la misma comunidad en la comunidad real pero sí pertenecen a la misma en la comunidad real. El rango de valores [-1,1]. Cuanto mayor sea el valor, más coherente será el resultado real con el real. Este indicador tiene un mayor grado de discriminación que el NMI.

En el caso del modularidad, cuando es optimizada se puede obtener un mejor resultado de división de la comunidad. Esta hace que los nodos en una misma comunidad estén más estrechamente conectados y a su vez ampliamente separados de otras comunidades, por lo que es un buen medidor de la fuerza de la comunidad; véase la fórmula general de la modularidad en el cap. 2.3.

### 1.3.1 Modularidad:

La modularidad es una medición cuantitativa de la densidad de las relaciones internas de una comunidad comparada con el número de relaciones existentes en la comunidad en general(Khan, Niazi 2017) ,o sea, es un indicador de qué tan óptima es la partición de un grafo en comunidades. Por tanto, la optimización de esta medida es una de las técnicas más utilizadas para hallar comunidades en un tiempo razonablemente rápido.
La ecuación de modularidad varía en dependencia de las condiciones que presenta el grafo que se está trabajando:

$$Q=\frac{1}{2m}\sum_{i,j}[A_{ij}-\frac{k_i k_j}{2m}]\quad \delta(\ C_i, C_j\ )$$

*Ecuación 3 Modularidad en redes no ponderadas*

Para la ecuación 3:
M es el número de enlaces, k el grado del vértice i o j, $C_i$ es la comunidad en que aparece el vértice i y $C_j$ la comunidad en que aparece el vértice j.

$$Q=\frac{1}{2W}\sum_{i,j}[A_{ij}-\frac{S_i S_j}{W}]\delta(C_i C_j)$$

*Ecuación 4 Modularidad en redes ponderadas*

Para la ecuación 4:
W es el peso total de los enlaces de la red y S es la fuerza del vértice i o j.

$$Q=\frac{1}{2m}\sum_{i,j}[A_{ij}-\frac{k_i^{out}k_j^{¿}}{m}]\delta(C_iC_j)$$

*Ecuación 5Modularidad en redes ponderadas y dirigidas*

En el caso de una comunidad superpuesta o solapada, que es aquella donde las comunidades pueden tener subcomunidades la obtención de la modularidad se obtiene:

$$Q=\frac{1}{2m}\sum_{ij}\frac{1}{O_iO_j}(A_{ij}-\frac{k_ik_j}{2m})\delta(C_iC_j)$$

*Ecuación 6 Modularidad en redes solapadas no ponderadas y no dirigidas*

$$Q=\frac{1}{m}\sum_{c=1}^{n_c}\sum_{i,j}[r_{ijc}A_{ijc}-S_{ijc}\frac{k_i^{out}k_j^{¿}}{m}]$$

*Ecuación 7 Modularidad en redes solapadas no ponderadas y dirigidas*

En el caso de la ecuación 6 Oi es la cantidad de nodos que tiene el módulo i, y en consecuente Oj es la cantidad de nodos que tiene el módulo j

En la ecuación 7 C es el índice que etiqueta cada comunidad, Rijc y Sijc representan las contribuciones a la suma correspondiente a la relación los nodos i y j en la red y el modelo nulo debido a la cantidad de afiliaciones de i y j.

Para resolver los problemas de las estructuras comunitarias de las redes (Liu et al. 2012) se ha definido una nueva medida cuantitativa denominada densidad de la modularidad (D) basada en la densidad de los subgrafos.

$$\frac{2\lambda L(V_i,V_j)-2(1-\lambda)L(V_i-\acute{V}_j)}{¿V_i\vee¿}$$

$$D_\lambda=\sum_{i=1}^{m}¿$$

*Ecuación 8 Densidad de la modularidad*

## 1.4 Panorama actual de la detección de comunidades

La detección de comunidades en la actualidad goza de gran atención, surgiendo nuevos intereses por parte de la comunidad científica en temes relacionados con redes sociales académicas, por ejemplo, se desarrolló un algoritmo de clasificación para responder a la interrogante: ¿Cuál es la línea de investigación más atractiva en la comunidad científica? (Wang, Han 2021).

El uso del deep learning ha propuesto modelos de aprendizaje profundo no supervisados para la detección de comunidades que permite extraer características de la red y usarlas en la división de la misma (Al-Andoli, Cheah, Tan 2021; Lopez Pinaya et al. 2020).

Otra aplicación notable de la detección de comunidades fue el desarrollo de un método que analizaba el comportamiento de comunicación entre nodos de una red mediante su IP, lo cual permitió determinar con precisión la estructura de redes a grandes escales, lo cual significa una herramienta provechosa para la gestión de la Seguridad (Zhang et al. 2022).
La aplicación de los algoritmos de detección de comunidades que mejores resultados arrojarían en la red creada teniendo en cuenta el trabajo que nos aborda es la aplicación de un algoritmo basado en nodo semilla como el Mux-Locid, como el aplicado por Hmimida y Kanawati en 2015 con redes híbridas que presentan relaciones entre autores, citas y sedes (Hmimida, Kanawati 2015b).

## 1.5 Metodología computacional

Con la naciente disciplina científica de analizar la exuberante cantidad de información existente se han propuesto nuevas metodologías ágiles que distan de las convencionalmente utilizadas para el desarrollo de software. Entre las principales actualmente se encuentran Knowledge Discovery in Databeses (KDD); SEMMA cuyas siglas en ingles significan: muestra, exploración, modificación, modelado, analisis y la tercera metodología es SCRIP DM cuyo nombre deriva de Cross Industry Standard Process for Data Mining (Gustavo Adolfo García Vélez 2018).

CRISP-DM es una metodología que se utiliza para el desarrollo de proyectos de ciencia de datos (Hotz 2018), la cual está dividida en 6 fases esenciales:

1. Comprensión del negocio.
2. Comprensión de los datos.
3. Preparación de los datos.
4. Modelación.
5. Evaluación.
6. Despliegue.

### 1.5.1 Comprensión del negocio

Esta fase se centra en el entendimiento de los objetivos y requisitos del proyecto y queda dividida en 4 tareas principales:

- Determinar los objetivos del negocio: Qué desea lograr el cliente
- Analizar la situación: Con qué recursos se cuenta, analizar riesgos y contingencias, realizar un análisis de costo.
- Determinar las metas de colección de datos: Además de definir los objetivos de la empresa, también debe definir qué aspecto tiene el éxito desde el punto de vista técnico de la minería de datos.
- Producir un plan para el proyecto: Seleccionar las tecnologías y herramientas.

### 1.5.2 Comprensión de los datos

La comprensión de los datos se enfoca en identificar, recopilar y analizar los conjuntos de datos con los que se trabajará para lograr los objetivos planteados. Esta fase también puede ser dividida en tres tareas:

- Recopilar los datos iniciales
- Describir los datos: Examinar los datos y documentar sus propiedades superficiales como el formato, la cantidad de datos obtenida.
- Explorar los datos: Visualizar los datos y hallar la relación entre ellos.
- Verificar la calidad de los datos: Qué tan limpios son los resultados de extracción de la información.

### 1.5.3 Preparación de los datos

En la fase de preparación de los datos se preparan los conjuntos de datos para su modelado, está compuesto por cinco tareas:

- Seleccionar los datos: Seleccionar cuáles serán los conjuntos de datos usados y documentar la justificación de tal decisión
- Limpiar los datos: Suele ser la tarea que más tarda en realizarse y es un paso fundamental para el éxito del proyecto, así se evita el ruido en los datos. Una práctica común es corregir, imputar o eliminar valores erróneos.
- Construir los nuevos datos: Derivar nuevos atributos que serán de ayuda, por ejemplo, derivar la masa corporal de una persona en altura y peso.
- Integrar los datos: Crear nuevos conjuntos de datos combinándolos desde múltiples fuentes.
- Dar formato a los datos: En ocasiones es necesario convertir valores para realizar operaciones matemáticas.

### 1.5.4 Modelado

Aquí se construirán y analizarán varios modelos basados en diferentes técnicas de modelado, está compuesta por cuatro tareas:

- Seleccionar la técnica de modelado: Determinar qué algoritmos serán utilizados.
- Generar un diseño de prueba: Dependiendo del acercamiento del modelo escogido puede ser necesaria una división de los datos para realizar pruebas o entrenamientos de los algoritmos.
- Construir modelo.
- Analizar modelo: Generalmente se ponen a prueba varios modelos y se comparan sus resultados mediante la interpretación basado en el conocimiento del campo.

La guía de esta metodología sugiere que se realicen varias iteraciones de cada modelo hasta estar muy seguros de que será suficientemente bueno para futuras iteraciones.

### 1.5.5 Evaluación

En esta fase se analiza qué modelo se ajusta mejor a la empresa y que hacer a continuación, para esto se enfoca en tres tareas:

- Evaluar los resultados de los modelos.
- Revisar el proceso: Revisar el trabajo realizado, buscar ausencia de algún paso y resumir los resultados.
- Determinar los pasos siguientes: A partir de las tareas anteriores se determina si es necesaria iterar nuevamente, si se puede implantar el modelo, o si se pueden iniciar nuevos proyectos.

### 1.5.6 Despliegue

Luego de terminar todo el proceso de desarrollo del proyecto la nueva fase se centra en como el interesado puede acceder a los resultados, la complejidad de esta fase puede variar en dependencia del proyecto y está dividida en cuatro tareas:

- Plan de despliegue:
- Plan de monitorización y mantenimiento:
- Producir un reporte final: El equipo del proyecto documenta un resumen del proyecto que puede incluir una presentación final de los resultados de la extracción de datos.
- Revisar el proyecto: Realiza una retrospectiva del proyecto sobre lo que ha ido bien, lo que podría haber sido mejor y cómo mejorar en el futuro.

### 1.5.7 Ventajas

Esta metodología se destaca por la posibilidad de reutilizar proyectos, por su independencia de la industria, su neutralidad con respecto a las herramientas y su enfoque en situaciones de negocio y análisis técnico, además de ser una metodología sin propietario

## 1.6 Entorno de Desarrollo Integrado

Jupyter Notebook es una aplicación en línea que le permite crear y compartir documentos que contienen código, ecuaciones, multimedia y texto. La capacidad de admitir 40 lenguajes de programación y diferentes tipos de recursos la convierte en una herramienta muy popular que le permite encontrar un conjunto muy completo de soporte, ejemplos y documentación en la web, lo que le permite crear nuevas aplicaciones y clases de funciones.

Jupyter Notebook está diseñado para facilitar la informática interactiva. Hay varias formas de usar Jupyter Notebooks en un entorno educativo. Con esta aplicación, se pueden generar libros estáticos con materiales de texto y ecuaciones, libros de trabajo para completar conjuntos de

práctica, informes, tareas, pruebas, aplicaciones y plataformas multimedia interactivas, plataformas de presentación y materiales de programación para el aula.

Beneficios de Jupyter Notebook:

- Código abierto
- Gratuito.
- Funciona en el navegador.
- Código en vivo (Live-Code)
- Diferentes opciones a la hora de exportar y compartir los resultados.
- Control de versiones.
- Permite colaboración (JupyterHub)

## 1.7 Lenguaje y módulos

Python 3 es un lenguaje de alto nivel multiparadigma, interpretado, multiplataforma y de tipado dinámico. (Amazon Web Services 2022)
Al ser un lenguaje interpretado se presentan ventajas:

- No se necesita compilar ahorrando tiempo en el desarrollo y prueba de la aplicación.
- El código fuente puede ser ejecutado en cualquier software siempre y cuando este disponga del intérprete.
- Su tipado dinámico hace un poco más cómodo el trabajo con las variables, ya que las variables son declaradas por su contenido y no por su contenedor, por lo que una variable puede ser reutilizada con otro tipo de dato en el futuro del código, además que no se necesita declarar el tipo de variable como en otros leguajes ahorrándonos tiempo.
- Cuenta con una biblioteca estándar con código reutilizable.
- Comunidad muy activa.
- Gran variedad de recursos educativos que se pueden encontrar fácilmente en muchas plataformas.

### 1.7.1 Módulos

Python tiene una amplia cantidad de módulos que ayudan a desarrollar aplicaciones con distintos propósitos. Esto trae consigo como ventajas que no es necesario tener instalados todos los módulos para el desarrollo de un producto, además de que al ser un lenguaje con el que cualquiera puede construir sus propios módulos y publicarlos siempre que el código sea eficiente se pueden utilizar dichos códigos ahorrando tiempo, para la utilización de estos módulos es necesario importarlos con la palabra clave **import** seguida del nombre del módulo o librería que se desea usar.

En el caso de la ciencia de datos hay módulos indispensables para trabajar, y en este trabajo serán utilizados:

- pandas
- matplotlib
- networkx
- networkx.algorithms.community
- serpapi

**pandas:**

Pandas es una librería de código abierto que brinda estructuras de datos de alto rendimiento y herramientas de análisis de datos. Para ello trabaja sobre el módulo NumPy.

NumPy es una librería que aporta una estructura de datos a bajo nivel que apoya a arreglos multidimensionales y un amplio rango de operaciones matemáticas sobre los mismos. Pandas tiene un mayor nivel de interfaz y aporta alineación simplificada de datos tabulares y una potente funcionalidad de series temporales.

La estructura clave en Pandas son los DataFrames, los cuales permiten almacenar y manipular los datos en estructuras bidimensionales los cuales pueden ser vistos como tablas y sobre estas provee características de conjuntos, por ejemplo, alineación de datos, estadísticas, agrupamiento, mezclas, concatenación de datos, etc.

**matplotlib:**

Matplotlib es una librería de código abierto que permite crear visualizaciones de datos. La visualización de datos es una etapa clave del análisis de datos. Después de haber recopilado, almacenado y analizado los datos es esencial transformarlos los resultados en informes y visualizaciones gráficas. Un diagrama es una forma más rápida para el humano de comprender de manera general las estadísticas.

Matplotlib se basa en elementos clave: Una “Figura” es una ilustración completa donde cada trazado de esa figura es llamado “eje”. El “plotting” consiste en crear una gráfica donde se utilizan los datos en formas de pares clave/valor los cuales se pueden interpretar como los ejes de las

abscisas y ordenadas. Después se utilizan funciones como “scatter”, “bar” y “pie” para crear el esquema con dichos datos.

Pyplot es un submódulo de matplotlib que se utilizará debido a que propone funciones sencillas para añadir a las figuras, tales como líneas, imágenes o textos a los ejes de un gráfico. Su interfaz es cómoda por lo que el uso de este módulo es regular en los desarrolladores del campo del análisis de datos.

**networkx:**

Networkx es una librería que se utiliza para la construcción y manipulación de estructuras gráficas complejas y proporciona además algunos algoritmos para analizar estos gráficos. Estos gráficos o grafos son estructuras compuestas por vértices (nodos), aristas y atributos opcionales. Los nodos representan los datos y las artistas las relaciones entre dos de estos nodos.

**serpapi:**

Serpapi es una API desarrollada por Google para investigadores que permite la extracción de datos de búsquedas en diferentes fuentes de Google.

### 1.7.2 JSON

JSON es un formato de intercambio de datos ligero que se hace fácil de entender por humanos tanto para escribir como leer y para las máquinas es fácil de analizar y generar. Su formato es basado en texto utilizando el lenguaje JavaScript, pero esto no hace que sea dependiente de dicho lenguaje, ya que usa convenciones familiares para los programadores de C, C++, C#, Java, JavaScript, Pearl, Python y otros. Razón por la que JSON se ha convertido en el formato de intercambio de información ideal en la actualidad.

JSON está construido en dos estructuras:

- Una colección de pares nombre-valor. En los lenguajes anteriormente mencionados esta estructura es vista de diferentes maneras: objetos, registros, estructuras, diccionarios, tabla hash, una lista o un arreglo.
- Una lista ordenada de valores, en la mayoría de lenguajes es vista como un arreglo, un vector, una lista, o una secuencia.

## 1.8 Pipeline de datos

Un pipeline de datos es una construcción lógica dividida en fases que definen un procesamiento de datos para su posterior análisis.

La creación de data pipelines es necesaria ya que no es recomendable analizar los datos en el mismo sistema en que son creados. El proceso de análisis es costoso computacionalmente, por lo que se separa para evitar perjudicar el rendimiento del servicio, de esta manera se tiene un sistema para la obtención y tratamiento de información y otro para su análisis.

**Pipeline ETL**

Los pipelines ETL se refiere a los procesos con las fases de extracción de los datos y su posterior transformación o filtrado y su carga en un sistema destino.

## 1.9 Conclusiones parciales

En la realización de este capítulo se investigó sobre las principales investigaciones que giran en torno al campo de la microfluídica y de los algoritmos de detección de comunidades, una vez visto los resultados de esta investigación se llegó a la conclusión que la información analizada brinda las herramientas y tecnologías necesarias que cumplen los requisitos necesarios para el cumplimiento de los objetivos planteados.

# 2 CAPÍTULO 2: DISEÑO DE LA PROPUESTA SOLUCIÓN

En el presente capítulo se describirá la propuesta de solución con un enfoque teórico. Además, se presenta la construcción de la red la implementación del data pipeline y posteriormente una explicación de los algoritmos utilizados, así como su pseudocódigo.

## 2.1 Descripción del contexto organizacional

Los desarrolladores buscaron en Google scholar publicaciones científicas relacionadas a la microfluídica con palabras claves y almacenarán los ficheros encontrados, así como sus metadatos, con esta bibliografía se construirá una red heterogénea conformada por autores y publicaciones como nodos y las relaciones entre ellos como las coautorías o cocitaciones. A partir de esta red se implementarán diferentes algoritmos de detección de comunidades y se monitorearán los resultados para compararlos y determinar si son útiles para crear comunidades científicas en el campo de la microfluídica, y si es el caso cuál de ellos obtiene mejores y más rápidos resultados.

## 2.2 Modelado de la propuesta solución

### 2.2.1 Propuesta de solución

El objetivo de la investigación es aplicar distintos algoritmos de detección de comunidades en una red compuesta por publicaciones y autores en el campo de la microfluídica y analizar los resultados obtenidos por los mismos. Para ello, mediante la utilización de los distintos módulos existentes de Python se creará un grafo que represente la red tomando los datos ofrecidos por una API de Google Scholar utilizando las palabras claves “computational microfluidics”. Una vez creada esta red se aplicarán los algoritmos de detección de comunidades y se graficará su resultado para luego ser evaluados calculando la modularidad de los resultados.

## 2.3 Modelados

### 2.3.1 Construcción de la red

Para construir la red se implementó un data pipeline y el proceso del mismo está representado en la figura 1. Este proceso comienza con la obtención de los datos con los que se va a trabajar y

almacenados, luego son revisados para estandarizarlos y realizar una limpieza de los mismos, y finalmente se construye la red con el conjunto de datos resultantes de dicha limpieza.

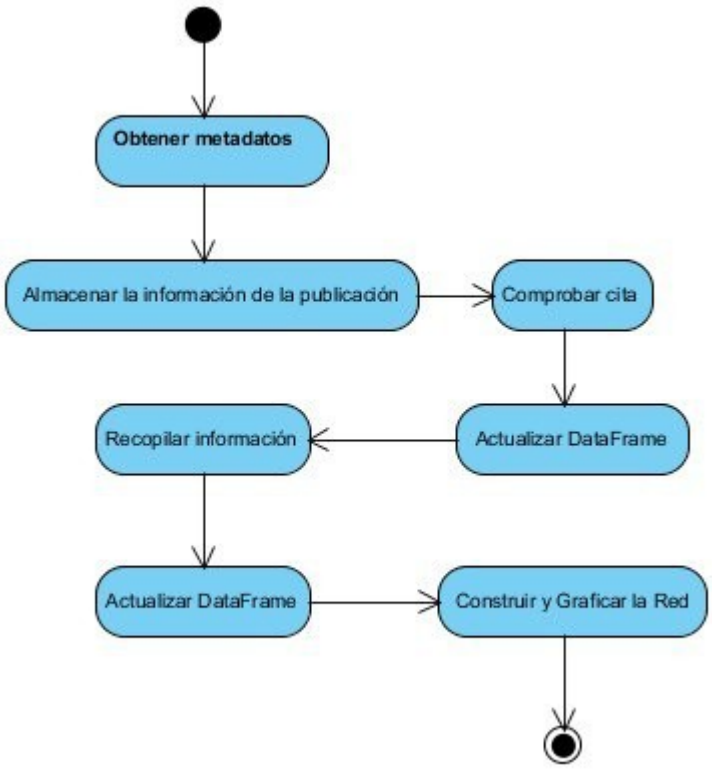



*Figura 1:Flujorama para la creación de la red*

Para la obtención de la información con que será construida la red se utilizó la API de Google Scholar con el siguiente código en Python:

```
from serpapi import GoogleSearch

params = {
  "q": "Coffee",
  "location": "Austin, Texas, United States",
  "hl": "en",
  "gl": "us",
  "google_domain": "google.com",
  "api_key": "secret_api_key"
}

search = GoogleSearch(params)
results = search.get_dict()
```

Los parámetros entregados al método GoogleSearch son definidos mediante un diccionario de Python con los parámetros de búsqueda, donde los más importantes son:

q: El valor de esta clave serán las palabras claves de la búsqueda que se realizará.
api_key: Ese parámetro es una llave única que tiene cada cuenta registrada en dicha API.

La ventaja de utilizar esta API es que ahorra tiempo en la construcción de un script que realice scrapping para la obtención de los datos necesarios para construir la red, proceso que se torna muy difícil por las políticas de cada sitio web.

La desventaja de la misma es que funciona con planes a pago, por lo que con un plan gratuito solo se pueden realizar hasta 100 búsquedas mensuales, por lo que para obtener una cantidad considerable de información habría que pagar o dedicar meses de utilización de la API solo para recopilar la información.

Los resultados de la búsqueda anterior fueron almacenados en un DataFrame de pandas y posteriormente guardado en un archivo csv para evitar pérdidas de la información mientras se analizaban los datos que contiene. Luego estos datos son comprobados para eliminar del DataFrame aquellas publicaciones que no son citadas, las cuales no aportarían información y los algoritmos verán como una comunidad única o la añadirán a una comunidad de manera aleatoria:

```
for index, row in df.iterrows():
  a=row['inline_links']
  if type(a) is str:
    a=ast.literal_eval(a)
  try:
    b=a['cited_by']
    c=b['serpapi_scholar_link']
  except:
    df.drop(index, inplace=True)
```

Finalmente es guardada esta última versión y construido el grafo con la librería Networkx.

### 2.3.2 Algoritmo de Propagación de etiquetas

El algoritmo de propagación de etiquetas es un algoritmo rápido para encontrar comunidades en un grafo, el cual se basa únicamente en la estructura del grafo, por lo que no requiere una función objetivo predefinida o información previa sobre las comunidades.

El funcionamiento del algoritmo es el siguiente:

- Primeramente, los nodos son inicializados con una etiqueta de comunidad única.
- Las etiquetas son propagadas por la red.
- En cada iteración del algoritmo, cada nodo es actualizado asignándole a su etiqueta el valor de la etiqueta a la que pertenecen el máximo número de nodos conectados al mismo; en caso de que exista un empate entre estos nodos adyacentes se le asigna el nuevo valor de forma aleatoria.
- El criterio de convergencia del algoritmo se cumple cuando todos los nodos tienen la etiqueta mayoritaria de sus vecinos, y el algoritmo se detiene cuando alcanza este criterio o llega a un número de iteraciones máxima definida por el usuario.

A modo de conclusión: Una vez aplicado el algoritmo, los grupos de nodos densamente conectados obtienen una etiqueta única rápidamente. Al final de la propagación quedarán pocas etiquetas y la mayoría habrán sido eliminadas.

Una característica del algoritmo es que se pueden asignar etiquetas preliminares a los nodos para reducir el rango de soluciones generadas, pero esto podría afectar la precisión en el caso de estudio que se abarca en este trabajo.

El pseudocódigo del algoritmo se muestra a continuación:

**Requiere:**

$G^0 = (V^0, E^0)$ : *Grafo inicial no dirigido donde* , $V^0$ *es el conjunto de nodos iniciales y* $E^0$ *el conjunto de relaciones en*

$Max_{iteraciones}$ : *Cantidad de iteraciones máximas* $(opcional)$

$\gamma$ : *Parámetro de resolución*

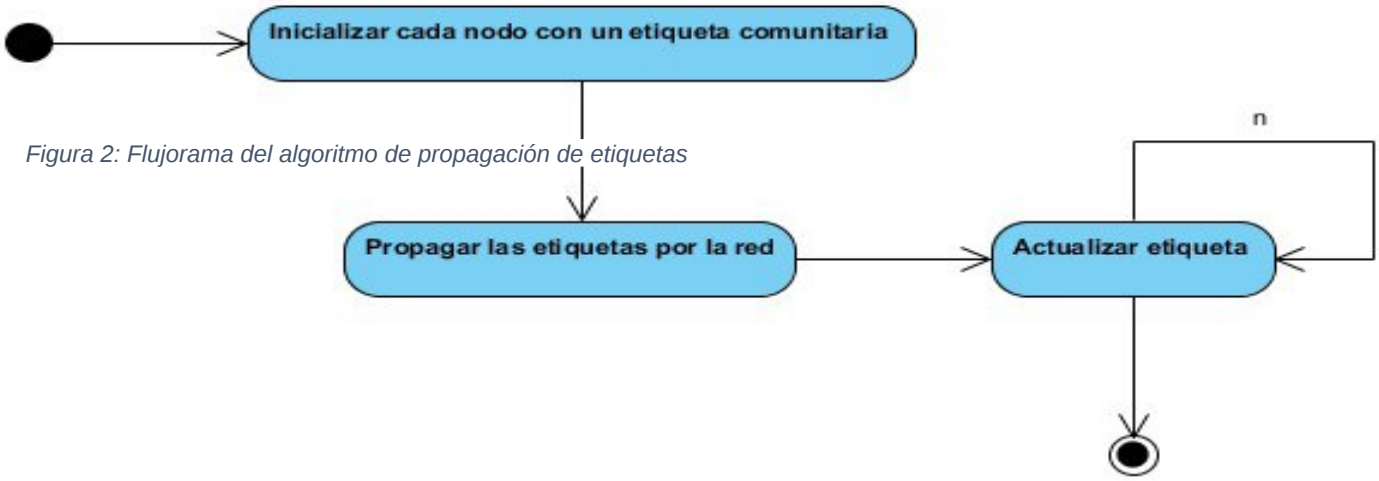



*Figura 2: Flujorama del algoritmo de propagación de etiquetas*

**Asegurarse:**

M : módulo resultante

$k_i$ : *número de vecinos con la etiqueta* $\lambda_i$

$v_i$ : *número total de nodos con la etiqueta* $\lambda_i$

**Pseudocódigo:**

1: k = 0 //número de iteración

2: // Atribuyendo una comunidad diferente para cada nodo.

3: para todo i en $V_0$ :

4: $M_i^0$ = {i}

5: terminar ciclo

6: //Hacer aleatorio el orden de los vértices

7: repetir:

8: para todo $u \in V^k$ :

9: para todo $i \in etiquetas(u)$:

10: si $k_i(u) - \gamma(v_i(u) - k_i(u)) > k_{i-1}(u) - \gamma(v_{i-1}(u) - k_i(u))$ entonces:

11: $M_u^k = M_u^k \setminus \{u\}; M'^k_u = M'^k_u \cup \{u\}$

12: terminar condicional

13: terminar ciclo

14: terminar ciclo

15: k = k+1

16: hasta que no exista un movimiento de los vértices o k < $max_{iteraciones}$

### 2.3.3 Algoritmo de Louvain

Louvain es un algoritmo no supervisado que no requiere conocer la cantidad o tamaño de las comunidades de antemano, el mismo se divide en dos fases:

- Optimización de la modularidad.
- Agregación de comunidades.

Ambas fases son ejecutadas iterativamente hasta que no hay más cambios en la red y se alcance el mayor nivel de modularidad.

**Optimización de la modularidad:**

Louvain ordenará aleatoriamente todos los nodos y luego, uno a uno, los cambiará de comunidad hasta que verifica que no hay un aumento significativo de la modularidad.

**Agregación de comunidades:**

Luego de terminar el primer paso, todos los nodos que pertenecen a la misma comunidad son mezclados en un solo nodo, las aristas entre los nuevos son la suma de los que conectan previamente nodos de las mismas comunidades diferentes. Este paso también genera bucles propios que son la suma de todos los enlaces dentro de una comunidad determinada, antes de ser colapsados en un nodo.

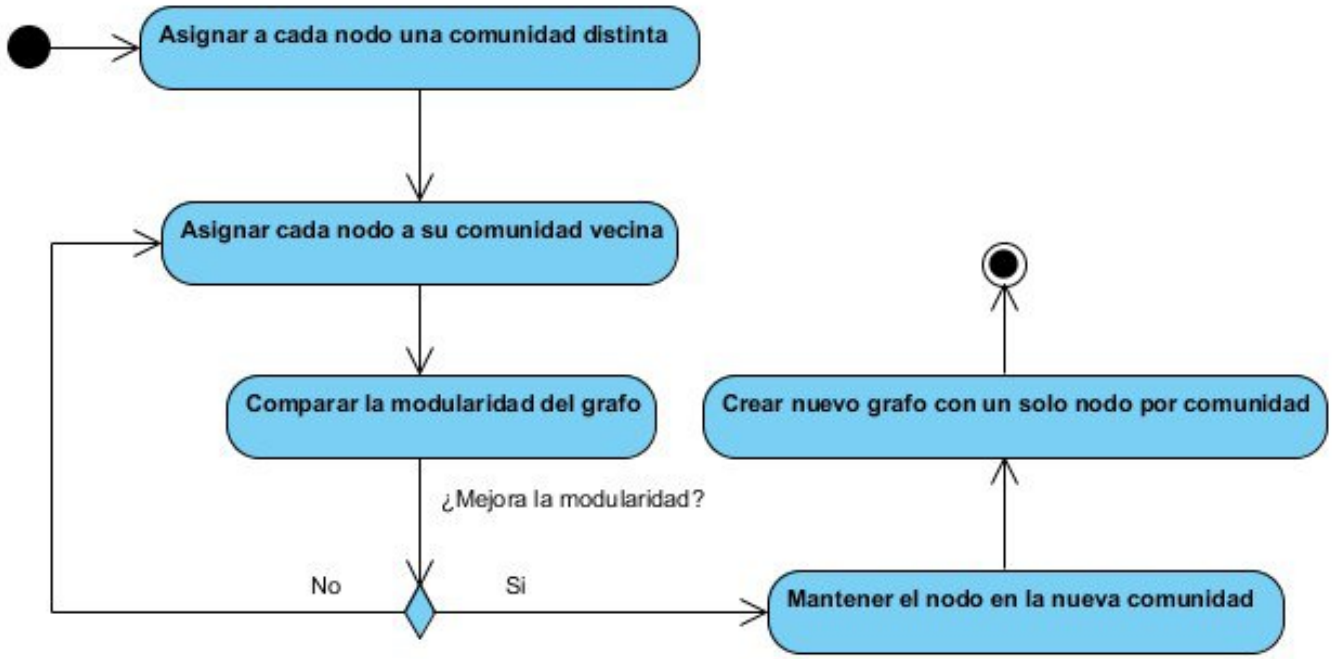



*Figura 3 Flujorama de algoritmo de Louvain*

El pseudocódigo para la implementación de este algoritmo es el siguiente:

**Requiere:**

$G^0 = (V^0, E^0)$: *Grafo no dirigido inicial donde* $V^0$ *es el conjunto inicial de nodos y* $E^0$ *es el conjunto inicial de aristas*.

$\varnothing$: *Umbral de mejora de la modularidad*

**Asegurar:**

M : Módulo resultante

Mod : Modularidad resultante

$\delta Mod$: *Variación de la modularidad*

$k_i$: *Suma del peso de todas las aristas que conectan al nodo i*

$k_{i,j}$: *Peso de las aristas que conectan al nodo i con el j*

1:m = $\sum k_{i,j}, (i, j) \in E^0$

2: k = 0 //número de iteración

3: Repetir:

4: //Asignando una comunidad a cada nodo:

5: para todo $i \in V^k$ :

6: $\sum_{i}^{k} ¿(i)$

7: terminar ciclo

8: Calcular $Mod_{nueva} = Mod(M)$

9: Repetir:

10: $Mod = Mod_{nueva}$

11: //Hacer aleatorios el orden de los nodos

12: para todo $i \in V^k$ :

13: mejor_comunidad = $M_i^k$

14: mejor_incremento = 0

15: para todo $M' \in C^k$:

16: $¿ M_i^k = M_i^k \}$

17: $\sum_{tot} M_i^k = \sum_{a \in M_i^k} k_a - k_i ; \sum_{tot} M'^k_i = \sum_{a \in M_i^k} k_a + k_i$

18: $\sum_{¿} ¿ \sum_{tot} M'^k_i - \sum k_{i,j}, (i,j) \in C_i^k \; y \; j \notin C_i^k$

19: $i, \in ¿ = \sum_{a \in M'^k_i} k_{i,a}$
$k_{¿}$

20: si $\delta Mod_{M_i^k \to M'^k_i} > mejor_{incremento}$:

21: mejor_incremento = $\delta Mod_{M_i^k \to M'^k_i}$

22: mejor_com = $M'^k_i$

23: $M'^k_i = M'^k_i \cup \{i\}$

24: si no:

25: $M_i^k = M_i^k \cup \{i\}$

26: terminar condicional

27: terminar ciclo

28: terminar ciclo

29: calcular $Mod_{nueva}$ = Mod(M)

30: Hasta que no halla movimiento en los nodos o $Mod_{nueva} - Mod < 0$

31: //Calcular la modularidad actualizada

32: Calcular $Mod_{nueva} = Mod(M)$

33: si $Mod_{nueva} - Mod < 0$ :

34: romper

35: terminar condicional

36: Mod = $Mod_{nueva}$

37: // unir las comunidades en un grafo

38: $V^{k+1} \leftarrow C^k$

39: $E^{k+1} \leftarrow e\left(C_u^k, C_v^k\right)$

40: $G^{k+1} = \left(V^{k+1}, E^{k+1}\right)$

41: k = k+1

42: hasta que se rompa

## 2.4 Conclusiones parciales

Luego de diseñada la propuesta de solución se llegó a la conclusión que los algoritmos seleccionados son los ideales para ser implementados ya que las condiciones de la red cumplen con los requisitos de ambos algoritmos, permitiendo un análisis de publicaciones sin necesidad de recurrir a proveedores de información que cobran por tal servicio o un extenso periodo de recopilación de información.

# 3 CAPÍTULO 3: PRUEBAS Y RESULTADOS DE LA INVESTIGACIÓN

**Introducción:**

En el presente capítulo se presentan los resultados tanto de la construcción de la red como de los algoritmos, además de un rápido análisis sobre la efectividad de dichos algoritmos llegando a una conclusión con respecto a los mismos. También recogerá un análisis sobre los nodos con mayor cantidad de relaciones, dígase, mas importantes llegando al final a una recomendación fundamental sobre este análisis.

## 3.1 Construcción de la red:

Debido a que se decidió utilizar la API de Google Scholar no se pueden realizar varias iteraciones de obtención de datos. Esto se debe a que dicha API funciona con un código único vinculado a una cuenta de Google donde la cantidad de resultados que se pueden obtener están limitados en dependencia de un plan pagado. El plan utilizado para desarrollar la propuesta de solución fue el gratuito con una limitación de 100 páginas de resultados.

Luego de ejecutado el código para la obtención de los datos y almacenados en un DataFrame de pandas queda el mismo con las siguientes columnas de valores:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81 entries, 0 to 80
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Unnamed: 0        81 non-null     int64
 1   Unnamed: 0.1      81 non-null     int64
 2   Unnamed: 0.1.1    81 non-null     int64
 3   position          81 non-null     int64
 4   title             81 non-null     object
 5   result_id         81 non-null     object
 6   link              81 non-null     object
 7   snippet           81 non-null     object
 8   publication_info  81 non-null     object
 9   resources         54 non-null     object
 10  inline_links      81 non-null     object
 11  type              23 non-null     object
dtypes: int64(4), object(8)
```

*Ilustración 4 Datos que aporta la API*

Donde quedan almacenados los metadatos de las búsquedas:

*Ilustración 5 Elementos reales del DataFrame*

| title | result_id | link | snippet | publication_info | resources | inline_links |
|---|---|---|---|---|---|---|
| Computational inertial microfluidics: A review | BrMcxlUdPgsJ | https://pubs.rsc.org/en/content/articlehtml/20... | ... In this review, we have summarized all compu... | {'summary': 'SR Bazaz, A Mashhadian, A Ehsani,... | [{'title': 'uts.edu.au', 'file_format': 'PDF',... | {'search_metadata': {'id': '6349b610654a8cc964... |
| Computational microfluidics for geosciences | JLEWIQjHjzYJ | https://www.frontiersin.org/articles/10.3389/f... | ... Our definition of computational microfluidic... | {'summary': 'C Soulaine, J Maes, S Roman - Fro... | [{'title': 'frontiersin.org', 'file_format': '... | {'search_metadata': {'id': '6349c02fc640d21b98... |
| Lattice Boltzmann method for microfluidics: mo... | UNDHHXxKH3oJ | https://link.springer.com/article/10.1007/s104... | ... Moreover, recent LBM applications in various... | {'summary': 'J Zhang - Microfluidics and Nanof... | [{'title': 'researchgate.net', 'file_format': ... | {'search_metadata': {'id': '6349c030f0adfb7a46... |

En la anterior figura se pueden evidenciar los atributos o propiedades que van a tener las publicaciones. De ellas las más importantes son:

“title”: Es el título de la publicacion científica
“result_id” : Es un identificador único para cada publicación (Será utilizado en la construcción de la red para identificar los nodos)
“publication_info” : Contiene los autores de dicha publicacion en formato de cita
“inline_links” : Contiene el identificador de las publicaciones que citan a la misma

El dataset tiene un total de 81 publicaciones, a partir de las cuales se construyo la red.
Una vez implementado la construccion del grafo queda con un total de 803 nodos:

```
G = nx.Graph()
for index,row in df.iterrows():

  sumary=ast.literal_eval(row[8])
  G.add_node(row[5],title=row[4])
  G.add_node(sumary['summary'])
  G.add_edge(row[5],sumary['summary'],color='b')

  aux=ast.literal_eval(row['inline_links'])
  for i in aux['organic_results']:
    try:
      G.nodes[i['result_id']]
    except:
      sumary=i['publication_info']['summary']
      G.add_node(i['result_id'],title=i['title'],sumary=sumary)
      G.add_edge(row[5],i['result_id'],color='r')
```

```
len(G)
```

```
803
```

*Ilustración 6 Cantidad de publicaciones finales representadas en el grafo*

Finalmente la graficacion de la red queda de la siguiente manera con 803 nodos y 728 aristas:

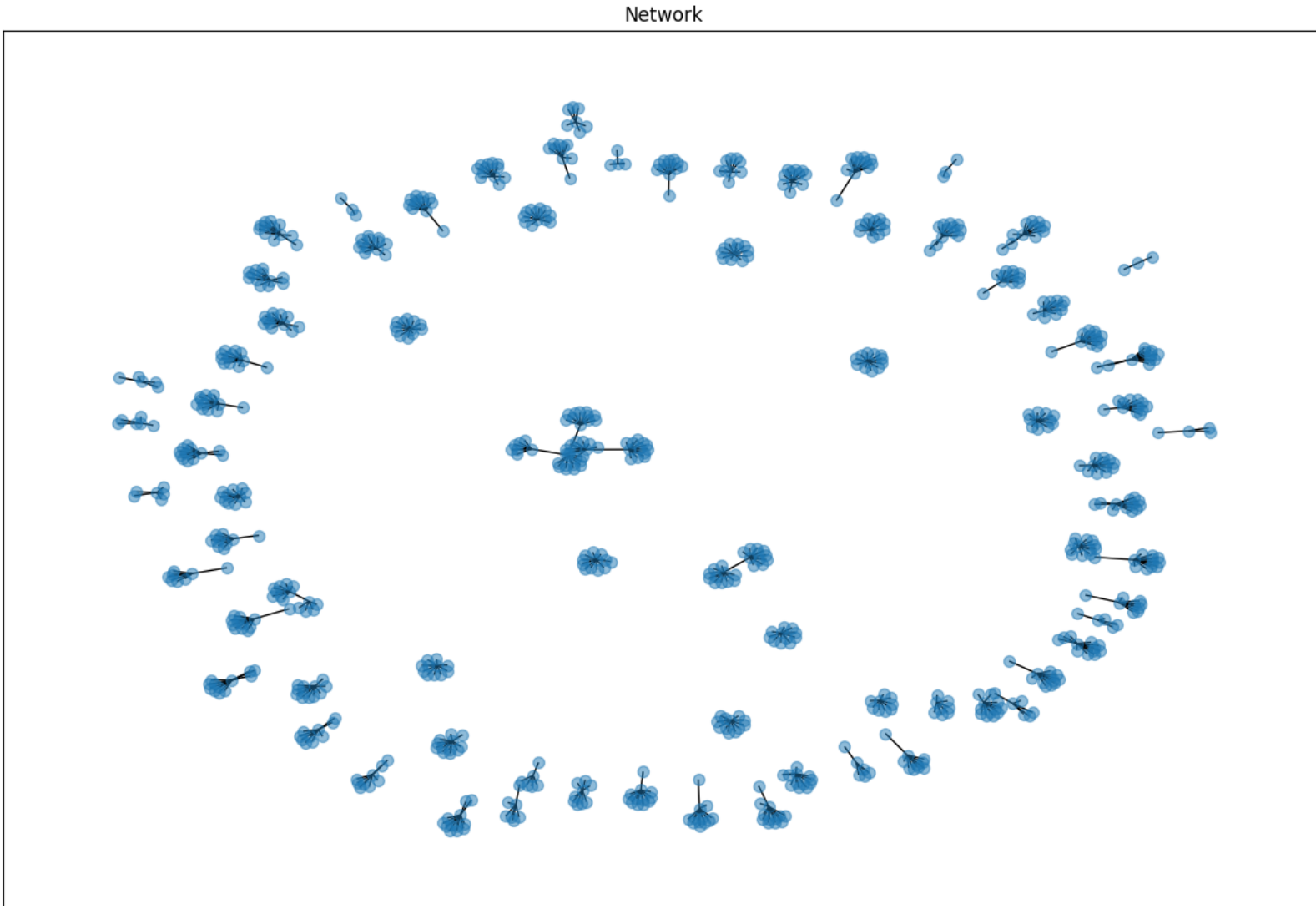
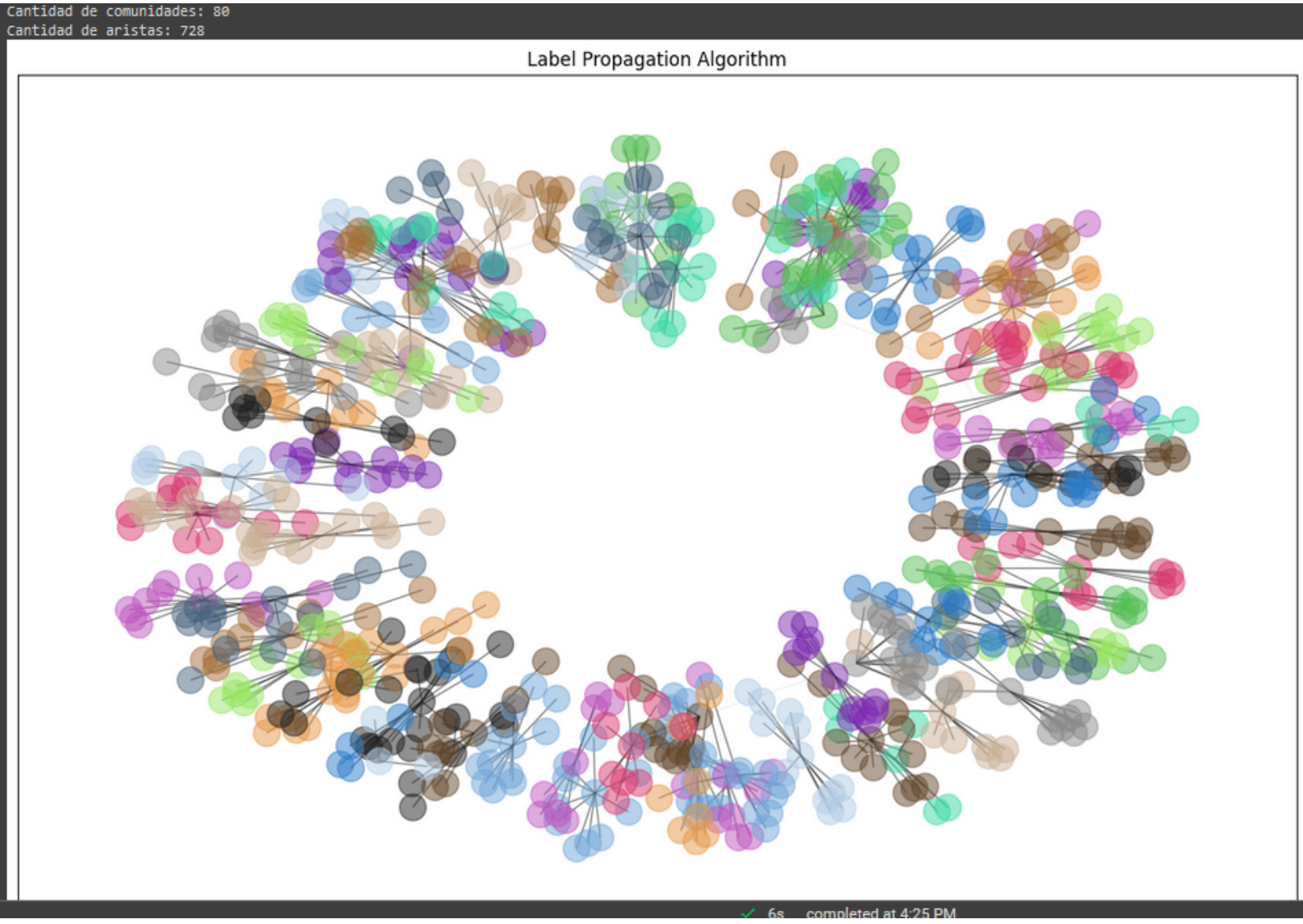



*Ilustración 7 representación gráfica de la red*

*Ilustración 8 Representación de manera aleatoria de la red*

## 3.2 Pruebas que se realizaran:

Para cada paso de este trabajo se decidió realizar varias corridas de los algoritmos implementados para ver la diferencia entre cada iteración y comparar ambos algoritmos, en el caso de la construcción de la red, no se pueden comparar iteraciones de obtención de datos debido a la limitante de la API en cuanto a cantidad de búsquedas permitidas, en caso de querer comparar dos o más iteraciones de esta socavación de información se debería adquirir uno de los




planes de Google, pero se realizará una prueba unitaria para com.

### 3.2.1 Algoritmo de porpagación de etiquetas:

Una vez aplicado el algoritmo de propagación de etiquetas queda la red de la siguiente manera:

*Ilustración 9 1ra iteración del algoritmo de propagación de etiquetas*

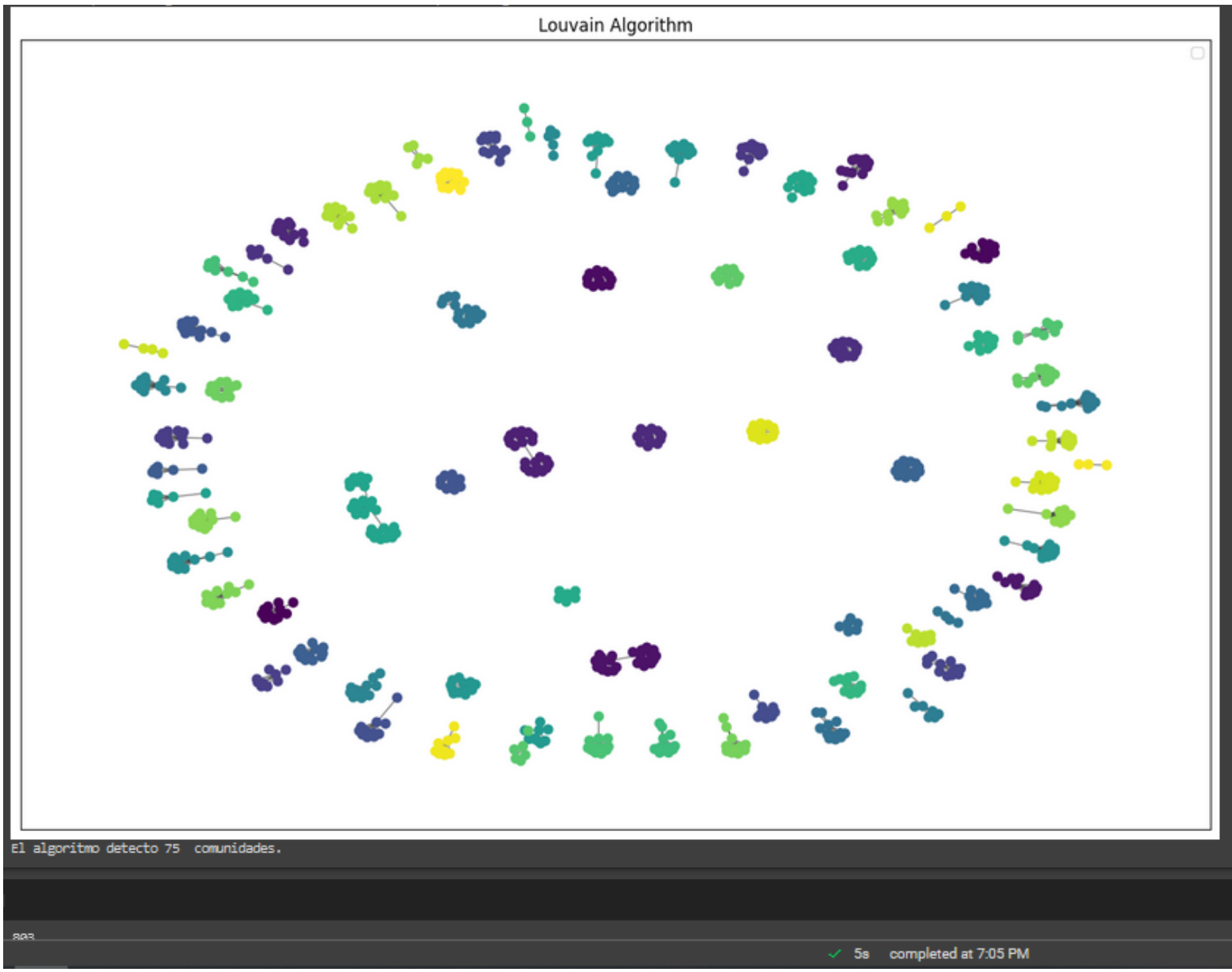



*Ilustración 10 2da iteración del algoritmo de propagación de etiquetas*

Como se puede evidenciar en cada iteración el algoritmo determina que existen 80 comunidades, y el resultado es el mismo incluyendo el tiempo de respuesta, así que se puede considerar estable el algoritmo para el caso de la red creada.

La modularidad de la red luego de aplicado el algoritmo es: 0.9795257970051926 el cual es un buen resultado y se mantiene estable en cada iteración del algoritmo.

### 3.2.2 Algoritmo de Louvain:

Al ser aplicado el algoritmo de Louvain se obtiene como resultado:

*Ilustración 11 1ra iteración del algoritmo de Louvain*

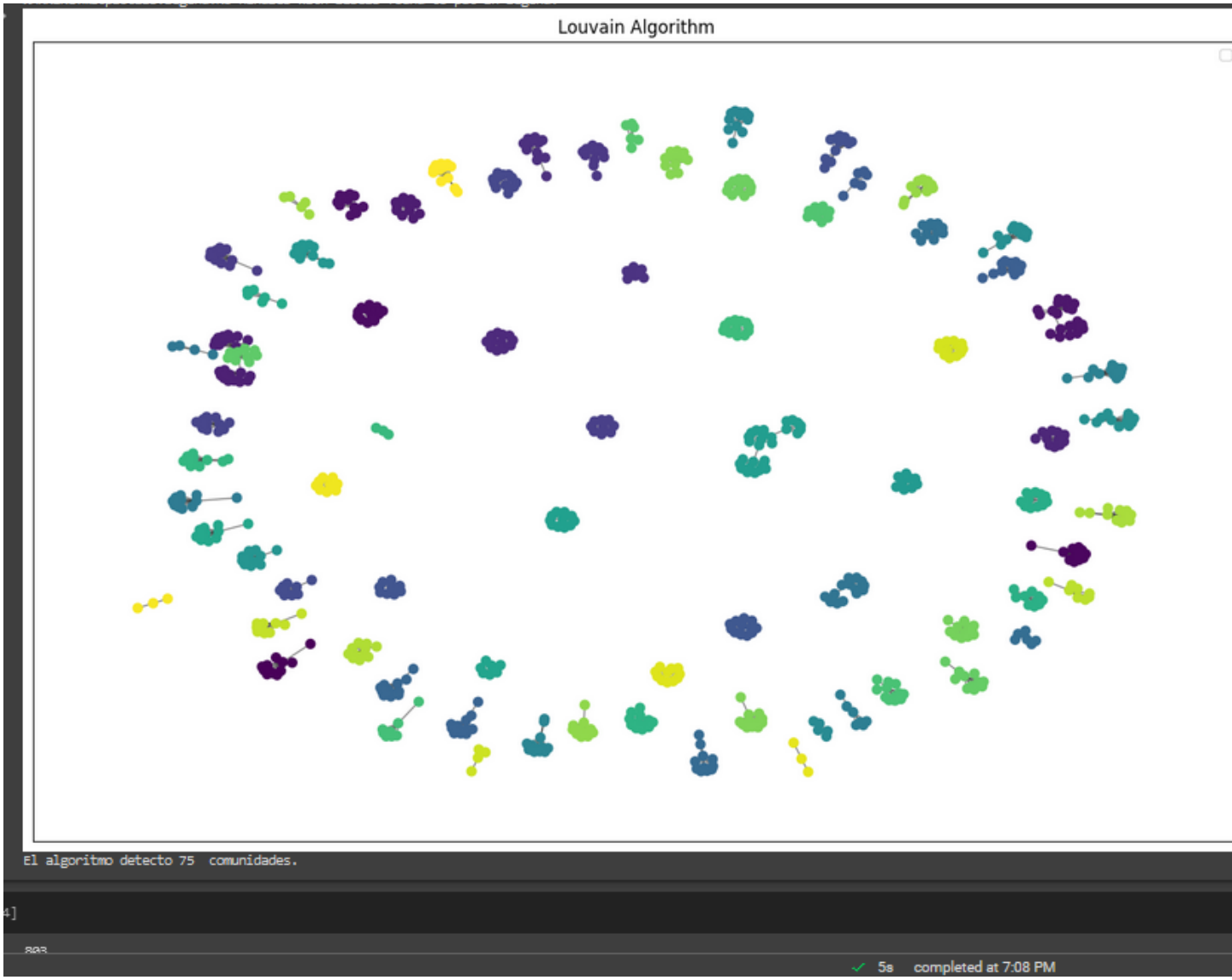



*Ilustración 12 2da iteración del algoritmo de Louvain*

Al igual que el algoritmo de propagación de etiquetas, el de Louvain se muestra estable y más rápido que el anterior, pero detecta menos comunidades, lo que nos interesa en este trabajo ya que se ajusta más a los nodos más importantes con una modularidad de 0.9844787767177878, superando al algoritmo de propagación de etiquetas en tiempo y eficiencia.

### 3.2.3 Prueba unitaria

Las pruebas unitarias son una forma de comprobar que un fragmento de código funciona correctamente. Es un proceso fundamental en el desarrollo de cualquier sistema a través de una metodología ágil que validan mediante pequeñas pruebas la lógica y el comportamiento de un objeto. A pesar de que toma tiempo desarrollar dichas pruebas, con ellas se detectan errores que

podrían afectar el rendimiento sin tener que llegar a etapas avanzadas del desarrollo de un proyecto (Yeeply 2022).

Una vez terminada la obtención y limpieza de los datos, los mismos son utilizados para la construcción de un grafo y de esta manera quedarían representados. Para comprobar que el código funciona correctamente y se añadieron los autores y publicaciones al grafo se realizó una prueba unitaria, donde la primera iteración de la prueba arrojó el resultado mostrado en la ilustración 13:

*Ilustración 13 Prueba unitaria*

Luego del análisis del error se llegó a la conclusión que se debía a un error con el entorno debido a que se usó Jupyter Notebook, de manera que se procedió a la corrección del código quedando finalmente resuelto el problema como es mostrado en la ilustración 14:

```
import unittest

class TestStrToBool(unittest.TestCase):

    def comprobacion(grafo,dataset):
      for index,row in dataset.iterrows():
        if row[5] in grafo.nodes():
          return True
        else: return False


if __name__ == '__main__':
    unittest.main()
```

```
E
======================================================================
ERROR: /root/ (unittest.loader._FailedTest)
----------------------------------------------------------------------
AttributeError: module '__main__' has no attribute '/root/'

----------------------------------------------------------------------
Ran 1 test in 0.002s

FAILED (errors=1)
An exception has occurred, use %tb to see the full traceback.

SystemExit: True

SEARCH STACK OVERFLOW
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:3334: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

```
import unittest

def comprobacion(G,df):
  try:
    for index,row in df.iterrows():
        if row[5] in G.nodes():
          pass
    return True
  except: return False



class Testcomprobacion(unittest.TestCase):

    def test_is_true(self):
        result = comprobacion(G,df)
        self.assertTrue(result)


if __name__ == '__main__':
  s = unittest.TestLoader().loadTestsFromTestCase(Testcomprobacion)
  unittest.TextTestRunner().run(s)
```

```
.
----------------------------------------------------------------------
Ran 1 test in 0.006s

OK
```

*Ilustración 14 Solución del error de la prueba unitaria*

## 3.3 Análisis de los nodos más influyentes:

Si se considera que el objeto de estudio de una red social es más influyente mientras mayor sea la cantidad de conexiones que tiene, como por ejemplo las personas más influyentes en una red social como Twitter; y se adapta al contexto que nos aborda, una publicación es más importante en el campo de la microfluídica mientras mayor sea la cantidad de publicaciones que citan a la misma, por eso se ha calculado cuales son los nodos más influyentes en la red creada que cumplan con el criterio de más de 10 citaciones, por supuesto que este criterio puede variar pero se ha tomado un número relativamente pequeño por las dimensiones de la red.

El resultado de dicho calculo resalta 47 publicaciones relevantes en el campo de la microfluídica y se muestran gráficamente en la siguiente figura:

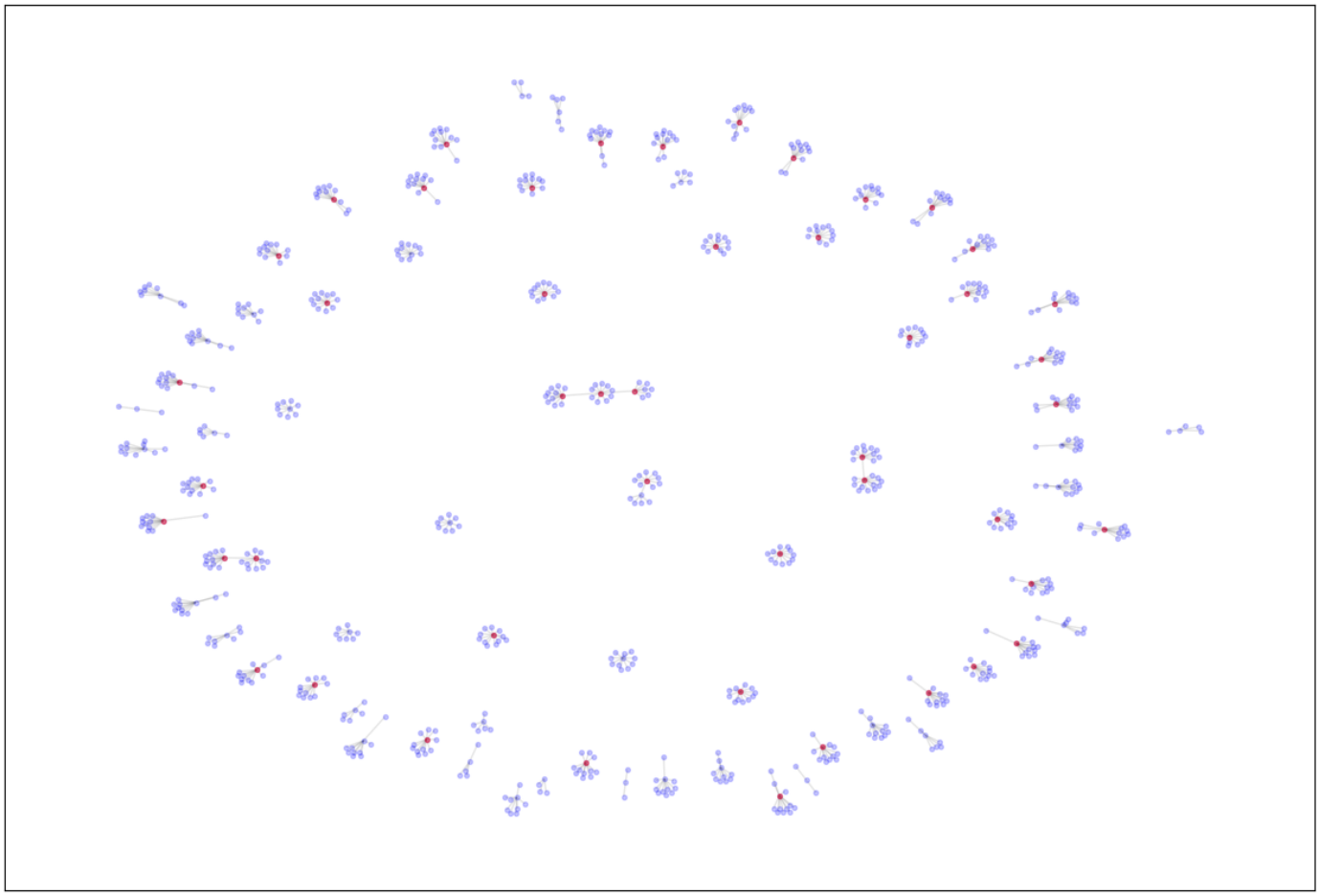


*Ilustración 15 Representación gráfica de los nodos de mayor peso*

Dichas publicaciones fueron almacenadas en una tabla (Poner el modo de lectura para poder ver las celdas enteras):

| | Titulo |
|---|---|
| **0** | Computational inertial microfluidics: A review |
| **1** | Computational microfluidics for geosciences |
| **2** | Lattice Boltzmann method for microfluidics: models and applications |
| **3** | Liquid flow in microchannels: experimental observations and computational analyses of microfluidics |
| **4** | Microfluidics of nano-drug delivery |
| **5** | Computational fluid dynamics (CFD) software tools for microfluidic applications–A case study |
| **6** | Numerical modeling of multiphase flows in microfluidics and micro process engineering: a review of r |
| **7** | Microfluidic cell culture system studies and computational fluid dynamics |
| **8** | Progress in computational microfluidics using TransAT |
| **9** | Mathematical modeling and computational analysis of centrifugal microfluidic platforms: A review |
| **10** | Improved fuel utilization in microfluidic fuel cells: A computational study |
| **11** | Computational design optimization for microfluidic magnetophoresis |
| **12** | Computational models in microfluidic bubble logic |
| **13** | A microfluidic device and computational platform for high-throughput live imaging of gene expression |
| **14** | Computational simulation of microfluidics, electrokinetics, and particle transport in biological mems d |
| **15** | Study of flow behaviors on single-cell manipulation and shear stress reduction in microfluidic chips u |
| **16** | Computational modeling of microfluidic fuel cells with flow-through porous electrodes |
| **17** | Combining molecular dynamics and lattice Boltzmann simulations: a hierarchical computational proto |
| **18** | Computational fluid dynamics analysis of microbubble formation in microfluidic flow-focusing devices |
| **19** | Computational analysis of enhanced magnetic bioseparation in microfluidic systems with flow-invasiv |
| **20** | Hemodynamic analysis for stenosis microfluidic model of thrombosis with refined computational fluid |
| **21** | Hemodynamic analysis for stenosis microfluidic model of thrombosis with refined computational fluid |
| **22** | Computational and functional evaluation of a microfluidic blood flow device |
| **23** | Numerical studies of shear-thinning droplet formation in a microfluidic T-junction using two-phase lev |
| **24** | Computational fluid dynamics in microfluidic systems |
| **25** | Two-phase computational modelling of a membraneless microfluidic fuel cell with a flow-through poro |
| **26** | Computational investigations of the mixing performance inside liquid slugs generated by a microfluid |
| **27** | Hemodynamics in the microcirculation and in microfluidics |

| | |
|---|---|
| **28** | … effect of factor VIII deficiencies and replacement and bypass therapies on thrombus form computational … |
| **29** | Microfluidics: fundamentals, devices, and applications |
| **30** | Computational fluid dynamics modelling of microfluidic channel for dielectrophoretic BioMEMs applic |
| **31** | Computational analysis of enhanced circulating tumour cell (ctc) separation in a microfluidic syst (DEP-MAP) … |
| **32** | Computational Microfluidic Channel for Separation of Escherichia coli from Blood-Cells |
| **33** | Mechanical disruption of mammalian cells in a microfluidic system and its numerical analysis based |
| **34** | Experimental and computational study of microfluidic flow-focusing generation of gelatin methacrylat |
| **35** | On the computational analysis of short mixing length planar split and recombine micromixers for mic |
| **36** | Fluid Mechanics for Chemical Engineers with Microfluidics and CFD. |
| **37** | Microfluidic-assisted preparation of PLGA nanoparticles for drug delivery purposes: experimental stu |
| **38** | Deep learning with microfluidics for biotechnology |
| **39** | Lateral migration of deformable particles in microfluidic channel flow of Newtonian and viscoelastic m |
| **40** | Computational study of pH-sensitive hydrogel-based microfluidic flow controllers |
| **41** | Experimental and computational evaluation of area selectively immobilized horseradish peroxidase i |
| **42** | Generalized numerical formulations for multi-physics microfluidics-type applications |
| **43** | Computational cell analysis for label-free detection of cell properties in a microfluidic laminar flow |
| **44** | A computational model of a microfluidic device to measure the dynamics of oxygen-dependent ATP r |
| **45** | Combining On-Chip Synthesis of a Focused Combinatorial Library with Computational Target Predic |
| **46** | Computational simulation of bio-microfluidic processes in integrated DNA biochips |
| **47** | A combined continuum/DSMC technique for multiscale analysis of microfluidic filters |

## 3.4 Conclusiones:

Luego de analizar los resultados obtenidos, se tiene como conclusión:

- La investigación sobre los referentes teóricos y metodológicos asociados con las investigaciones en el campo de la microfluídica y los algoritmos de detección de comunidades permitió decidir cómo crear la red utilizada y qué algoritmos implementar para validar la utilización de los mismos en futuras investigaciones.
- El algoritmo de propagación de etiquetas a pesar de tener una buena modularidad y buen tiempo de respuesta, no crea buenas particiones, ya que las comunidades que identifica son los resultados encontrados en una primera iteración, por lo que realizar una búsqueda en Google Scholar y ver cada resultado por separado tendría el mismo resultado.
- El algoritmo de Louvain es un poco más eficiente que el de propagación de etiquetas pero aún la cantidad de comunidades es cercano a la cantidad de publicaciones encontradas en la búsqueda inicial.
- Realizar una búsqueda de los nodos con mayor peso en esta red daría buenos resultados en cuanto a las publicaciones más relevantes, ya que se reducen la búsqueda de publicaciones a consultar en futuras investigaciones.

### 3.4.1 Recomendaciones:

Luego de analizados los resultados de este capítulo se tienen en cuenta varias recomendaciones:

- Con el análisis del último punto del capítulo se recomienda probar la red con un algoritmo basado en nodos semilla, ya que se espera tener resultados con buena modularidad y comunidades más precisas, basado en que el análisis de los nodos más influyentes logro reducir casi a la mitad las posibles comunidades.
- Seguir utilizando la API de Google Scholar debido a la buena organización de los datos, no obstante investigar la existencia de otras fuentes gratuitas de información académica pretendiendo encontrar nuevos rasgos de las publicaciones.
- Ampliar la red ya sea con mayor cantidad de resultados, como realizar búsquedas con otras palabras claves para comprobar que los algoritmos sigan siendo eficientes con un mayor número de nodos.
- Tener en cuenta una investigación analizando estadísticamente tanto a los autores de las publicaciones como la relación entre co-autores.

# 4 REFERENCIAS


1. AL-ANDOLI, Mohammed, CHEAH, Wooi Ping y TAN, Shing Chiang, 2021. Deep learning-based community detection in complex networks with network partitioning and reduction of trainable parameters. *Journal of Ambient Intelligence and Humanized Computing*. febrero 2021. Vol. 12, no. 2, pp. 2527-2545. DOI 10.1007/s12652-020-02389-x.

2. AMAZON WEB SERVICES, 2022. ¿Qué es Python? | Guía de Python para principiantes de la nube | AWS. *Amazon Web Services, Inc.* en línea. 2022. [Accedido 14 noviembre 2022]. Recuperado a partir de: https://aws.amazon.com/es/what-is/python/

3. BERLINGERIO, Michele, COSCIA, Michele y GIANNOTTI, Fosca, 2011. Finding and Characterizing Communities in Multidimensional Networks. En: *2011 International Conference on Advances in Social Networks Analysis and Mining*. en línea. Kaohsiung City, Taiwan: IEEE. julio 2011. pp. 490-494. [Accedido 25 junio 2022]. ISBN 978-1-61284-758-0. DOI 10.1109/ASONAM.2011.104.

4. BERROCAL, José L Alonso, FIGUEROLA, Carlos G y MEDRANO, José Federico, sin fecha. RECOLECCIÓN, DETECCIÓN DE COMUNIDADES Y VISUALIZACIÓN DE INFORMACIÓN WEB. . pp. 13.

5. BLEI, David M, 2003. Latent Dirichlet Allocation. . 2003. pp. 30.

6. CORDIS, 2020. La microfluídica es el futuro de la agricultura con análisis de suelo «in situ» | MobiLab Project | Results in brief | H2020 | CORDIS | European Commission. en línea. 2020. [Accedido 17 junio 2022]. Recuperado a partir de: https://cordis.europa.eu/article/id/421573-microfluidics-is-the-future-of-agriculture-with-on-site-soil-analysis/es

7. D. FERNÁNDEZ RIVAS, 2011. MICROFLUIDOS: NUEVAS FRONTERAS. en línea. 28 abril 2011. [Accedido 14 noviembre 2022]. Recuperado a partir de: http://www.revistacubanadefisica.org/RCFextradata/OldFiles/2011/vol.28-No.1/RCF-28-1-2011-60.pdf

8. DHUMAL, Amit y KAMDE, Pravin, 2015. Survey on Community Detection in Online Social Networks. *International Journal of Computer Applications*. 18 julio 2015. Vol. 121, no. 9, pp. 35-41. DOI 10.5120/21571-4609.

9. DIVAKARMURTHY, Pramod y MENEZES, Ronaldo, 2013. The Effect of Citations to Collaboration Networks. En: MENEZES, Ronaldo, EVSUKOFF, Alexandre y GONZÁLEZ, Marta C. (eds.), *Complex Networks*. en línea. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 177-185. Studies in Computational Intelligence. [Accedido 9 junio 2022]. ISBN 978-3-642-30286-2.

10. DUCH, Jordi y ARENAS, Alex, 2005. Community detection in complex networks using extremal optimization. *Physical Review E*. 24 agosto 2005. Vol. 72, no. 2, pp. 027104. DOI 10.1103/PhysRevE.72.027104.

11. FORTUNATO, Santo, 2010. Community detection in graphs. *Physics Reports*. febrero 2010. Vol. 486, no. 3-5, pp. 75-174. DOI 10.1016/j.physrep.2009.11.002. arXiv:0906.0612 [cond-mat, physics:physics, q-bio]

12. FORTUNATO, Santo y BARTHÉLEMY, Marc, 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*. 2 enero 2007. Vol. 104, no. 1, pp. 36-41. DOI 10.1073/pnas.0605965104.

13. GALVEZ, Carmen, 2018. Análisis de co-palabras aplicado a los artículos muy citados en Biblioteconomía y Ciencias de la Información (2007-2017). *Transinformação*. 1 diciembre 2018. Vol. 30, pp. 277-286. DOI 10.1590/2318-08892018000300001.

14. GONZÁLEZ, Jorge Dettmer, 2019. Análisis de Redes Sociales (ARS): Estado del arte del caso mexicano. *Espacio Abierto*. 2019. Vol. 28, no. 3, pp. 5-24.

15. GRANMA, 2022. Crean hidrogel capaz de sanar heridas con mayor rápidez › Ciencia › Granma - Órgano oficial del PCC. en línea. 2022. [Accedido 9 junio 2022]. Recuperado a partir de: https://www.granma.cu/ciencia/2015-06-06/crean-hidrogel-capaz-de-sanar-heridas-con-mayor-rapidez

16. GUIMERÀ, R., MARTA, S. P. y AMARAL, L. A., 2007. *Identificación de módulos en redes bipartitas y dirigidas. Physical Review E Statistical Nonlinear & Soft Matter Physics,*. . 2007. 76(2): 036102

17. GULÍN-GONZÁLEZ, E, PÉREZ-REYES, C M, TORRES-PUPO, C, COSTA-MARRERO, Y y NAVAS-CONYEDO, 2014. CARACTERIZACIÓN COMPUTACIONAL DE LA DIFUSIÓN EN UN SISTEMA UNIDIMENSIONAL DE PARTÍCULAS INTERACTUANTES EN RÉGIMEN HIDRODINÁMICO. . 2014. Vol. 31, no. 1, pp. 4.

18. GUPTA, Manish, AGGARWAL, Charu C., HAN, Jiawei y SUN, Yizhou, 2011. Evolutionary Clustering and Analysis of Bibliographic Networks. En: *2011 International Conference on Advances in Social Networks Analysis and Mining*. en línea. Kaohsiung City, Taiwan: IEEE. julio 2011. pp. 63-70. [Accedido 27 junio 2022]. ISBN 978-1-61284-758-0. DOI 10.1109/ASONAM.2011.12.

19. GUSTAVO ADOLFO GARCÍA VÉLEZ, 2018. *ACIÓN DE LA METODOLOGÍA CRISP-DM® A LA RECOLECCIÓN Y ANÁLISIS DE DATOS GEORREFERENCIADOS DESDE TWITTER*. en línea. Recuperado a partir de: https://repository.unimilitar.edu.co/bitstream/handle/10654/20099/GarciaVelezGustavoAdolfo2018.pdf;jsessionid=4BEE8632B53DECB7881ED246E8514F2A?sequence=3

20. HMIMIDA, Manel y KANAWATI, Rushed, 2015a. *Community detection in multiplex networks: A seed-centric approach*. . 2015.

21. HMIMIDA, Manel y KANAWATI, Rushed, 2015b. Community detection in multiplex networks: A seed-centric approach. *Networks and Heterogeneous Media*. 1 marzo 2015. Vol. 10, pp. 71-85. DOI 10.3934/nhm.2015.10.71.

22. HOTZ, Nick, 2018. What is CRISP DM? *Data Science Process Alliance*. en línea. 10 septiembre 2018. [Accedido 17 noviembre 2022]. Recuperado a partir de: https://www.datascience-pm.com/crisp-dm-2/

23. HUGHES, B. D., 1996. *Random walks and random environments*.

24. J. GULÍN-GONZÁLEZ y E. NAVAS-CONYEDO, 2017. HYDRODYNAMICAL CHARACTERIZATION OF RED BLOOD CELLS INTERACTIONS IN A HIGH CONFINEMENT REGIME. A COMPUTATIONAL STUDY. en línea. 25 octubre 2017. Recuperado a partir de:

http://www.revistacubanadefisica.org/RCFextradata/OldFiles/2018/Vol.35_No.1E/RCF_35_E12.pdf

25. JI, Ming, HAN, Jiawei y DANILEVSKY, Marina, 2011. Ranking-based classification of heterogeneous information networks. En: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*. en línea. San Diego, California, USA: ACM Press. 2011. pp. 1298. [Accedido 27 junio 2022]. ISBN 978-1-4503-0813-7. DOI 10.1145/2020408.2020603.

26. JIANG, Lu, GULÍN-GONZÁLEZ, Jorge, CONYEDO, Edisel Navas y CHEN, Yunwei, 2021. Methods of community detection in hybrid network applied to the academic network analysis. . 2021. Vol. 15, pp. 32.

27. JOLLIFFE, I. T., 2002. *Principal Component Analysis*. en línea. New York: Springer-Verlag. [Accedido 20 junio 2022]. Springer Series in Statistics. ISBN 978-0-387-95442-4.

28. KANAWATI, R. Licod, 2011. Licod: identificación de líderes para la detección de comunidades en redes complejas. . Nueva Jersey: IEEE. 2011.

29. KHAN, Bisma S y NIAZI, Muaz A, 2017. Network Community Detection: A Review and Visual Survey. . 3 agosto 2017. pp. 39.

30. LANCICHINETTI, Andrea, FORTUNATO, Santo y KERTÉSZ, János, 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*. 10 marzo 2009. Vol. 11, no. 3, pp. 033015. DOI 10.1088/1367-2630/11/3/033015.

31. LEE, Daniel D. y SEUNG, H. Sebastian, 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*. octubre 1999. Vol. 401, no. 6755, pp. 788-791. DOI 10.1038/44565.

32. LI, J., SUN, P.Y. y MAO, Q.R., 2018. Detección de comunidades basada en la fusión de rutas en redes de información heterogéneas. En: . Nueva Jersey: IEEE. 2018.

33. LI, Lin, FAN, Kefeng, ZHANG, Zhiyong y XIA, Zhengmin, 2016. COMMUNITY DETECTION ALGORITHM BASED ON LOCAL EXPANSION K-MEANS. *Neural Network World*. 2016. Vol. 26, no. 6, pp. 589-605. DOI 10.14311/NNW.2016.26.034.

34. LIU, Jin Xia, ZENG, Jian Chao, XUE, Yao Wen y WANG, Ying, 2012. Quantitative Function for Community Detection. *Advanced Materials Research*. enero 2012. Vol. 433-440, pp. 6441-6446. DOI 10.4028/www.scientific.net/AMR.433-440.6441.

35. LIU, Jun, WANG, Jiangzhou y LIU, Binghui, 2020. Community Detection of Multi-Layer Attributed Networks via Penalized Alternating Factorization. *Mathematics*. 13 febrero 2020. Vol. 8, no. 2, pp. 239. DOI 10.3390/math8020239.

36. LIU, T.L., GONG, M. M. y TAO, D. C., 2016. Factorización de matrices no negativas de cono grande. . 2016.

37. LIU, Wei y CHEN, Ling, 2013. Community detection in disease-gene network based on principal component analysis. *Tsinghua Science and Technology*. octubre 2013. Vol. 18, no. 5, pp. 454-461. DOI 10.1109/TST.2013.6616519.

38. LIU, Xin, LIU, Weichu, MURATA, Tsuyoshi y WAKITA, Ken, 2014. A FRAMEWORK FOR COMMUNITY DETECTION IN HETEROGENEOUS MULTI-RELATIONAL NETWORKS.

*Advances in Complex Systems*. noviembre 2014. Vol. 17, no. 06, pp. 1450018. DOI 10.1142/S0219525914500180.

39. LOPEZ PINAYA, Walter Hugo, VIEIRA, Sandra, GARCIA-DIAS, Rafael y MECHELLI, Andrea, 2020. Autoencoders. En: *Machine Learning*. en línea. Elsevier. pp. 193-208. [Accedido 22 agosto 2022]. ISBN 978-0-12-815739-8.

40. M. E. J. NEWMAN REVIEWED, 2003. The Structure and Function of Complex Networks. *SIAM Review*. 2003. Vol. 45, no. 2, pp. 167-256.

41. MACROPOL, Kathy y SINGH, Ambuj, 2010. Scalable discovery of best clusters on large graphs. *Proceedings of the VLDB Endowment*. septiembre 2010. Vol. 3, no. 1-2, pp. 693-702. DOI 10.14778/1920841.1920930.

42. MEI, Q.Z., CAI, D. y ZHANG, D., 2018. Modelado de temas con regularización de redes. En: . New York: ACM press. 2018.

43. MENG, Qinxue, TAFAVOGH, Siamak y KENNEDY, Paul J., 2014. Community detection on heterogeneous networks by multiple semantic-path clustering. En: *2014 6th International Conference on Computational Aspects of Social Networks*. en línea. Porto, Portugal: IEEE. julio 2014. pp. 7-12. [Accedido 9 julio 2022]. ISBN 978-1-4799-5940-2. DOI 10.1109/CASoN.2014.6920424.

44. MENG, Xiaofeng, SHI, Chuan, LI, Yitong, ZHANG, Lei y WU, Bin, 2014. Relevance Measure in Large-Scale Heterogeneous Networks. En: CHEN, Lei, JIA, Yan, SELLIS, Timos y LIU, Guanfeng (eds.), *Web Technologies and Applications*. en línea. Cham: Springer International Publishing. pp. 636-643. Lecture Notes in Computer Science. [Accedido 5 julio 2022]. ISBN 978-3-319-11115-5.

45. MIKA, S., RATSCH, G., WESTON, J., SCHOLKOPF, B. y MULLERS, K.R., 1999. Fisher discriminant analysis with kernels. En: *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468)*. en línea. Madison, WI, USA: IEEE. 1999. pp. 41-48. [Accedido 9 julio 2022]. ISBN 978-0-7803-5673-3. DOI 10.1109/NNSP.1999.788121.

46. MURATA, Tsuyoshi y IKEYA, Tomoyuki, 2010. A NEW MODULARITY FOR DETECTING ONE-TO-MANY CORRESPONDENCE OF COMMUNITIES IN BIPARTITE NETWORKS. *Advances in Complex Systems*. febrero 2010. Vol. 13, no. 01, pp. 19-31. DOI 10.1142/S0219525910002402.

47. NEWMAN, M. E. J. y GIRVAN, M., 2004. Finding and evaluating community structure in networks. *Physical Review E*. 26 febrero 2004. Vol. 69, no. 2, pp. 026113. DOI 10.1103/PhysRevE.69.026113.

48. NGUYEN, Nam-Trung, WERELEY, Steven T. y SHAEGH, Seyed Ali Mousavi, 2019. *Fundamentals and Applications of Microfluidics, Third Edition*. Artech House. ISBN 978-1-63081-365-9. Google-Books-ID: h3iFDwAAQBAJ

49. NICOSIA, V, MANGIONI, G, CARCHIOLO, V y MALGERI, M, 2009. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*. 16 marzo 2009. Vol. 2009, no. 03, pp. P03024. DOI 10.1088/1742-5468/2009/03/P03024.

50. ORTEGA, Maruja y MEZA, Oscar, 1993. *Grafos y algoritmos*. Caracas: Equinoccio. ISBN 978-980-237-072-6.

51. ORTIZ MUÑOZ, Ernesto y HIDALGO DELGADO, Yusniel, 2016. Detección de comunidades a partir de redes de coautoría en grafos RDF. *Revista Cubana de Información en Ciencias de la Salud*. marzo 2016. Vol. 27, no. 1, pp. 90-99.

52. PAPADOPOULOS, Symeon, KOMPATSIARIS, Yiannis y VAKALI, Athena, 2010. A Graph-Based Clustering Scheme for Identifying Related Tags in Folksonomies. En: BACH PEDERSEN, Torben, MOHANIA, Mukesh K. y TJOA, A. Min (eds.), *Data Warehousing and Knowledge Discovery*. Berlin, Heidelberg: Springer. 2010. pp. 65-76. Lecture Notes in Computer Science. ISBN 978-3-642-15105-7. DOI 10.1007/978-3-642-15105-7_6.

53. PARTHASARATHY, Dhruv, SHAH, Devavrat y ZAMAN, Tauhid, 2019. *Leaders, Followers, and Community Detection*. en línea. 22 septiembre 2019. arXiv. arXiv:1011.0774. [Accedido 9 junio 2022]. Recuperado a partir de: http://arxiv.org/abs/1011.0774arXiv:1011.0774 [physics, stat]

54. PONCE ORDÓÑEZ, Jéssica, PINO, Ariosto, MORETA, Orlando y SAMANIEGO MENA, Eduardo, 2019. Caracterización de factores que influyen en la baja producción científica de las universidades usando análisis de redes sociales. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*. 2 enero 2019. Vol. 17, pp. https://www.proquest.com/docview/2195121667?pq-origsite=gscholar&fromopenview=true.

55. QIU, Changhe, CHEN, Wei, WANG, Tengjiao y LEI, Kai, 2015. Overlapping Community Detection in Directed Heterogeneous Social Network. En: DONG, Xin Luna, YU, Xiaohui, LI, Jian y SUN, Yizhou (eds.), *Web-Age Information Management*. en línea. Cham: Springer International Publishing. pp. 490-493. Lecture Notes in Computer Science. [Accedido 28 junio 2022]. ISBN 978-3-319-21041-4.

56. QUEUPIL, Juan Pablo y MONTECINOS, Carmen, 2020. El Liderazgo Distribuido para la Mejora Educativa: Análisis de Redes Sociales en Departamentos de Escuelas Secundarias Chilenas. *REICE: Revista Iberoamericana sobre Calidad, Eficacia y Cambio en Educación*. 2020. Vol. 18, no. 2, pp. 97-114.

57. RUIZ-SCHUCLOPER, José, 2009. *Reconocimiento Lógico Combinatorio de Patrones. Teoría y Aplicaciones (2009).pdf*. . 13 septiembre 2009.

58. SANTOS, Jorge M. y EMBRECHTS, Mark, 2009. On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification. En: ALIPPI, Cesare, POLYCARPOU, Marios, PANAYIOTOU, Christos y ELLINAS, Georgios (eds.), *Artificial Neural Networks – ICANN 2009*. en línea. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 175-184. Lecture Notes in Computer Science. [Accedido 22 agosto 2022]. ISBN 978-3-642-04276-8.

59. SHI, Chuan, KONG, Xiangnan, YU, Philip S., XIE, Sihong y WU, Bin, 2012. Relevance search in heterogeneous networks. En: *Proceedings of the 15th International Conference on Extending Database Technology - EDBT '12*. en línea. Berlin, Germany: ACM Press. 2012. pp. 180. [Accedido 5 julio 2022]. ISBN 978-1-4503-0790-1. DOI 10.1145/2247596.2247618.

60. SHI, Chuan, LI, Yitong, ZHANG, Jiawei, SUN, Yizhou y YU, Philip S., 2017. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*. 1 enero 2017. Vol. 29, no. 1, pp. 17-37. DOI 10.1109/TKDE.2016.2598561.

61. SÓSOL-FERNÁNDEZ, R. E., MARÍN-LIZÁRRAGA, V. M., ROSALES-CRUZALEY, E. y LAPIZCO-ENCINAS, B. H., 2012. Análisis de células en dispositivos microfluídicos. *Revista mexicana de ingeniería química*. agosto 2012. Vol. 11, no. 2, pp. 227-248.

62. SUN, Yizhou, HAN, Jiawei, YAN, Xifeng, YU, Philip S. y WU, Tianyi, 2011. PathSim: meta path-based top-K similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*. agosto 2011. Vol. 4, no. 11, pp. 992-1003. DOI 10.14778/3402707.3402736.

63. SUN, Yizhou, HAN, Jiawei y YU, Yintao, 2009. Ranking-based clustering of heterogeneous information networks with star network schema. En: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*. en línea. Paris, France: ACM Press. 2009. pp. 797. [Accedido 27 junio 2022]. ISBN 978-1-60558-495-9. DOI 10.1145/1557019.1557107.

64. SUN, Yizhou, HAN, Jiawei, ZHAO, Peixiang, YIN, Zhijun, CHENG, Hong y WU, Tianyi, 2009. RankClus: integrating clustering with ranking for heterogeneous information network analysis. En: *Proceedings of the 12th International Conference on Extending Database Technology Advances in Database Technology - EDBT '09*. en línea. Saint Petersburg, Russia: ACM Press. 2009. pp. 565. [Accedido 27 junio 2022]. ISBN 978-1-60558-422-5. DOI 10.1145/1516360.1516426.

65. SUN, Yizhou, NORICK, Brandon, HAN, Jiawei, YAN, Xifeng, YU, Philip S. y YU, Xiao, 2013. PathSelClus: Integrating Meta-Path Selection with User-Guided Object Clustering in Heterogeneous Information Networks. *ACM Transactions on Knowledge Discovery from Data*. septiembre 2013. Vol. 7, no. 3, pp. 1-23. DOI 10.1145/2500492.

66. TANG, Lei, WANG, Xufei y LIU, Huan, 2009. Uncoverning Groups via Heterogeneous Interaction Analysis. En: *2009 Ninth IEEE International Conference on Data Mining*. en línea. Miami Beach, FL, USA: IEEE. diciembre 2009. pp. 503-512. [Accedido 5 julio 2022]. ISBN 978-1-4244-5242-2. DOI 10.1109/ICDM.2009.20.

67. Una plataforma microfluídica para la obtención de nanomedicinas para tratar cáncer, sin fecha. *Fundacion Argentina de Nanotecnologia*. en línea. [Accedido 9 junio 2022]. Recuperado a partir de: https://www.fan.org.ar/noticias/una-plataforma-microfluidica-para-la-obtencion-de-nanomedicinas-para-tratar-cancer/

68. WANG, Ran, SHI, Chuan, YU, Philip S. y WU, Bin, 2013. Integrating Clustering and Ranking on Hybrid Heterogeneous Information Network. En: PEI, Jian, TSENG, Vincent S., CAO, Longbing, MOTODA, Hiroshi y XU, Guandong (eds.), *Advances in Knowledge Discovery and Data Mining*. en línea. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 583-594. Lecture Notes in Computer Science. [Accedido 9 junio 2022]. ISBN 978-3-642-37452-4.

69. WANG, Yakun y HAN, Xiaodong, 2021. Attractive community detection in academic social network. *Journal of Computational Science*. abril 2021. Vol. 51, pp. 101331. DOI 10.1016/j.jocs.2021.101331.

70. WARZ, María Paz Contreras y VERDUGO, Juan Alejandro Zúñiga, sin fecha. DETECCIÓN DE COMUNIDADES EN REDES COMPLEJAS O GRAFOS USANDO METAHEURÍSTICAS. . pp. 44.

71. WASSERMAN, Stanley y FAUST, Katherine, 1994. *Social Network Analysis: Methods and Applications*. en línea. 1. Cambridge University Press. [Accedido 7 junio 2022]. ISBN 978-0-521-38707-1.

72. YAKOUBI, Z. y KANAWATI, R. Licod, 2014. *Un algoritmo impulsado por el líder para la detección de comunidades en redes complejas*. . 2014. Vietnam Journal of Computer Science.

73. YEEPLY, 2022. ▷ ¿Qué son las pruebas unitarias y cómo llevar una a cabo? *Yeeply*. en línea. 6 septiembre 2022. [Accedido 25 noviembre 2022]. Recuperado a partir de: https://www.yeeply.com/blog/que-son-pruebas-unitarias/

74. YUAN, Peiyan, WANG, Wei y SONG, Mingyang, 2016. Detecting Overlapping Community Structures with PCA Technology and Member Index. En: *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications*. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 18 junio 2016. pp. 121-125. MobiMedia '16. ISBN 978-1-63190-104-1.

75. ZHANG, Shuzhuang, ZHANG, Yaning, ZHOU, Min y PENG, Lizhi, 2022. Community detection based on similarities of communication behavior in IP networks. *Journal of Ambient Intelligence and Humanized Computing*. marzo 2022. Vol. 13, no. 3, pp. 1451-1461. DOI 10.1007/s12652-020-02681-w.

76. ZHANG, Xianchao, LI, Haixin, LIANG, Wenxin y LUO, Jiebo, 2016. Multi-type Co-clustering of General Heterogeneous Information Networks via Nonnegative Matrix Tri-Factorization. En: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. diciembre 2016. pp. 1353-1358. DOI 10.1109/ICDM.2016.0185.