UNIVERSIDAD
REY JUAN CARLOS

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA TELECOMUNICACIÓN

Curso Académico 2020/2021

Trabajo Fin de Grado

# MODELOS ESCALABLES PARA ANÁLISIS DE REDES SOCIALES CON APACHE SPARK

Autor : Lydia Garrido Muñoz

Tutor : José Felipe Ortega Soto

# Trabajo Fin de Grado

Modelos Escalables para Análisis de Redes Sociales con Apache Spark

**Autor :** Lydia Garrido Muñoz

**Tutor :** José Felipe Ortega Soto

La defensa del presente Proyecto Fin de Carrera se realizó el día de de 20XX, siendo calificada por el siguiente tribunal:

**Presidente:**

**Secretario:**

**Vocal:**

y habiendo obtenido la siguiente calificación:

**Calificación:**

Fuenlabrada, a de de 20XX

*Dedicado a*

*mi familia / mi abuelo / mi abuela*

# Agradecimientos

# Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- £De qué va este proyecto? £Cuál es su objetivo principal?
- £Cómo se ha realizado? £Qué tecnologías están involucradas?
- £En qué contexto se ha realizado el proyecto? £Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.

# Summary

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.

# Índice general

# Índice de figuras

# Capítulo 1

# Introducción

## 1.1. Planteamiento del problema

Es cada vez más habitual encontrar datos e información acerca de grafos que representen sistemas de diversa índole.

Los grafos se pueden analizar mediante el llamado Análisis de Redes Sociales (*Social Network Analysis*) [1], [1, 2].

En el contexto digital, las redes sociales son plataformas que permiten la interacción entre las personas que se encuentran dentro de ellas. Dentro de las redes sociales hay varios ámbitos dependiendo del objetivo al que esté enfocado: redes de relaciones, como pueden ser Instagram, Facebook y Twitter, redes profesionales, como LinkedIn, y redes de entretenimiento, como Youtube o Pinterest.

Esta tecnología de análisis ha ido aumentando su popularidad debido a las herramientas de gran utilidad que ofrece para conectar datos de múltiples ámbitos. Por ejemplo, las redes sociales utilizan la tecnología de grafos para que se pueda representar como nodos y enlaces entre los distintos personajes que forman la red. Como se puede observar en la figura 1.1, cada nodo representa una persona y cada arista muestra la relación entre dos nodos. Analizando dichos enlaces, las redes sociales consiguen descubrir patrones para ofrecer a sus perfiles recomendaciones y experiencias personalizadas.

Otro ejemplo de utilidad en las redes sociales consistiría en la búsqueda de cuentas falsas. Actualmente, muchas marcas están empezando a orientar sus estrategias de marketing a perfiles

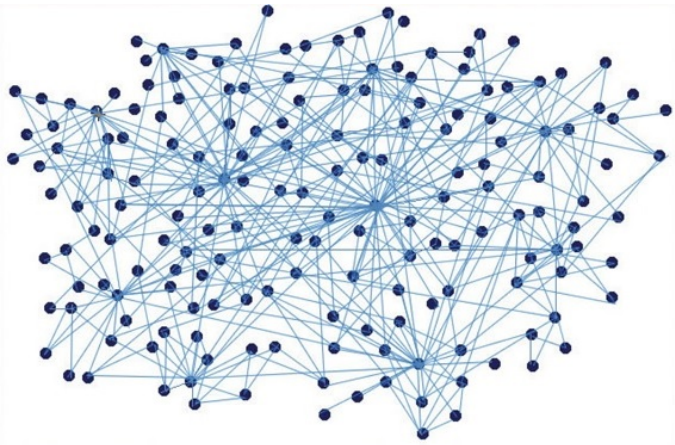

Figura 1.1: Grafo redes sociales. Extraido de [1].

populares en redes sociales. Gracias a los grafos pueden descubrir qué porcentaje de las cuentas que siguen a dichos perfiles son falsas.

Por otro lado, la tecnología de grafos también puede servir de ayuda para hacer seguimiento de contactos en una pandemia como la del COVID-19. Se pueden construir mapas de rastreo a partir de enlaces entre nodos. Si se detecta un caso positivo se pueden llevar a cabo acciones preventivas y descubrir la posible cadena de contagio gracias a las aristas que conectan a la persona contagiada con otros miembros de la red.

Puesto que los grafos tienen un gran tamaño y son cada vez más díficiles de analizar, se ha empezado a utilizar Spark para permitir escalado de operaciones manteniendo la misma eficiencia y GraphX para procesamiento de grafos.

## 1.2. Objetivos del proyecto

### 1.2.1. Objetivo general

El objetivo general consiste en evaluar la capacidad de GraphX para analizar conjuntos de datos escalables de gran tamaño.

### 1.2.2. Objetivos específicos

- Aprender Spark y GraphX.

- Realizar una prueba con un dataset pequeño.
- Realizar una prueba con un dataset de gran tamaño.

## 1.3. Estructura de la memoria

- En el capítulo 1 se hace una introducción al proyecto, presentando el contexto previo. Se exponen los objetivos que se van a llevar a cabo y se muestra la estructura de la memoria.
- En el capítulo 2 se presentan las tecnologías con las que se ha realizado el proyecto.
- A continuación se presentan los datasets utilizados para los experimentos del proyecto en el capítulo 3.
- En el capítulo 4 se comentan los experimentos realizados durante el proyecto.
- Por último se presentan las conclusiones en el capítulo 5. Dicho capitulo está formado por el resumen de los resultados, la estimación de esfuerzos, una breve explicación de las asignaturas de la carrera que han inspirado el proyecto y posteriormente las líneas futuras que se pueden llevar a cabo.

# Capítulo 2

# Estado del arte

En este capítulo se presentan las tecnologías fundamentales con las que se ha llevado a cabo este TFG. En primer lugar se presentan los conceptos fundamentales de la ciencia de análisis de redes. A continuación, se lleva a cabo una introducción a Apache Spark y posteriormente se explica en qué consiste GraphX.

## 2.1. Análisis de redes

El análisis de redes sociales es el proceso de investigación de las estructuras sociales mediante la teoría de grafos y el uso de redes. El objetivo del análisis de redes sociales es comprender una comunidad llevando a cabo un mapa de las relaciones que forman la red y, posteriormente, extraer los individuos clave, los grupos, también denominados componentes, y las conexiones entre estos. Una red se puede resumir como un número de nodos que están conectados por enlaces. Generalmente, en el análisis de redes sociales, los nodos son personas y los enlaces son las conexiones sociales entre ellas.

El análisis de redes sociales empezó de forma teórica con el trabajo de los primeros sociólogos, como Georg Simmel y Émile Durkheim, los cuales realizaron escritos sobre la importancia de estudiar los patrones sociales que conectan a las personas. Desde principios del siglo XX se utiliza el concepto de *redes sociales* para definir organizaciones complejas de relaciones entre individuos de sistemas sociales a todas las escalas.

Frente a las diferentes herramientas existentes para el análisis de redes sociales, los grafos presentan múltiples ventajas. Los grafos proporcionan una manera más eficiente de trabajar con

conceptos abstractos como las relaciones y las interacciones. A su vez, ofrecen una forma visual e intuitiva de entender estos conceptos. También consituyen una base natural para realizar un análisis de las relaciones en un contexto social.

Otra de las ventajas de la teoría de grafos, aplicada al análisis de las redes sociales. es la gran variedad de ámbitos en los cuales se puede aplicar. Por ejemplo, la Agencia de Seguridad Nacional de Estados Unidos hace uso de este tipo de análisis sobre redes terroristas y otros tipos de células que pueden ser peligrosas para la seguridad nacional, de este modo pueden determinar la estructura de estas y encontrar a sus respectivos líderes. Esto permite a los militares llevar a cabo ataques contra los activos al mando para poder interrumpir el funcionamiento de la red.

La necesidad de analizar e interpretar los datos para extraer conocimiento útil de ellos y el crecimiento incesante del volumen de información han provocado que el uso de sistemas distribuidos sea esencial en el análisis de redes a gran escala. Con este modelo logramos que cada nodo pueda ser tratado de manera independiente al resto. Además, conseguimos una mayor tolerancia a fallos, ya que al perder un nodo la información se encontrará en otro. También consta de mucha importancia que el sistema cuente con escalabilidad en sus diferentes facetas:

- **Escalabilidad de información**
  Consiste en la capacidad de extraer información de relevancia en grandes grafos.

- **Escalabilidad visual**
  Reside en la capacidad que poseen las herramientas de visualización de mostrar de forma óptima los datos extraídos de grandes grafos.

- **Escalabilidad analítica**
  Consiste en la capacidad de los algoritmos matemáticos de procesar eficazmente conjuntos de datos de gran tamaño.

- **Escalabilidad de software**
  Capacidad de un software de manejar óptimamente grandes cantidades de datos.

La escalabilidad puede ser de dos tipos: vertical u horizontal. El primero sucede cuando se agregan más recursos a un nodo, mientras que el segundo tipo se da cuando se añade un nuevo nodo a la red.

## 2.2. Apache Spark

Spark surgió a raíz de la aceptación de que las tecnologías anteriores utilizadas para el análisis de datos eran ineficientes para los trabajos de computación iterativa o interactiva. Los objetivos que se querían conseguir con este nuevo framework era lograr una carga de trabajo unificada, almacenar en memoria los resultados intermedios entre el mapa interactivo e iteravito para reducir tiempos de cómputo, que fuera muy tolerante a fallas y que permitirera ofrecer APIs disponibles en varios modelos de programación y fácilmente manejables.

Apache Spark es reconocido por reemplazar todos los motores separados de procesamiento en una pila de componentes unificada encargada de varias cargas de trabajo a partir de un único motor distribuido y rápido. Como se puede observar en la figura 2.1 Spark ofrece cuatro bibliotecas para múltiples cargas de trabajo: Spark SQL, Spark MLlib, Spark Structured Streaming y GraphX; y está disponible en los entornos de programación: SQL, Python, Scala, Java y R.

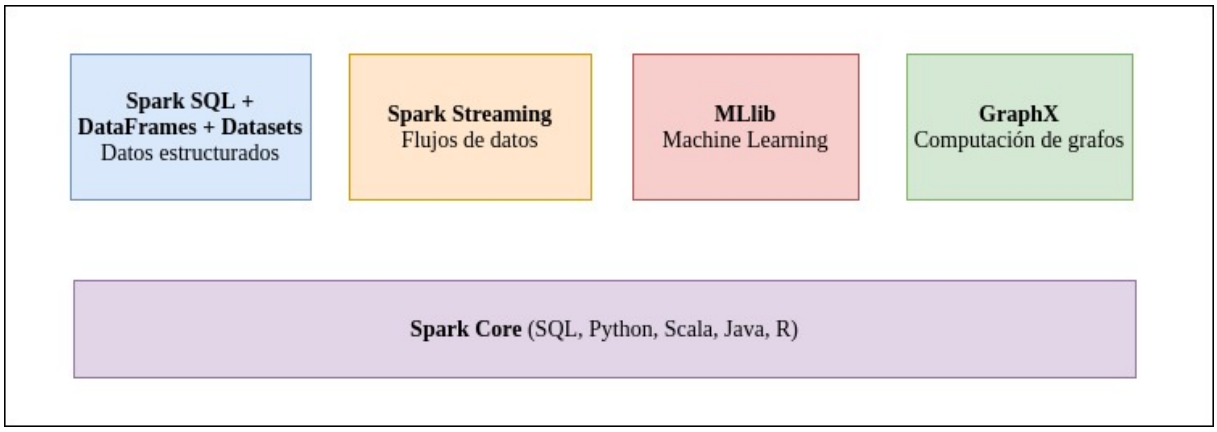


Figura 2.1: Componentes Apache Spark. Extraido de [2]

- **Spark SQL, DataFrames y Datasets**

  Este módulo está pensado para trabajar con datos estructurados. Permite leer datos almacenados en formatos de archivo como CSV, JSON, Parquet, XML (con bibliotecas adicionales), etc. También permite recoger datos de tablas Hive. Otra de las funcionalidades de este módulo es combinar consultas tipo SQL con operaciones programadas en Spark sobre datasets.

- **Spark Streaming**

  Esta biblioteca está orientada al análisis de datos en streaming manteniendo la misma escabilidad, fiabilidad y tolerancia a fallos. Permite que se puedan combinar en tiempo real tanto los datos estáticos como los datos en streaming. Admite diferentes fuentes de datos como Apache Kafka y otras fuentes de streaming.

- **MLlib**

  MLlib es una biblioteca que contiene algoritmos populares de aprendizaje automático (ML) construidos sobre APIs de alto nivel basadas en DataFrame para construir modelos en Spark. Se divide en dos partes: spark.mllib, que contiene los APIs basadas en RDD, y spark.ml, que es el API basado en DataFrame.

## 2.3. GraphX

GraphX es una biblioteca para la representación de grafos e implementa la computación paralela. Introduce Graph, un multigrafo dirigido en el que es posible asignar propiedades a cada enlace o vértice. También incluye una creciente colección de algoritmos y constructores de grafos para simplificar las tareas de análisis de grafos, y una variante optimizada de la API Pregel de Google, que es un sistema para el procesamiento de gráficos a gran escala. El operador Pregel en GraphX es una abstracción de mensajería paralela sincróna restringida a la topología del grafo a gran escala. Pregel se ejecuta en un serie de superpasos, en los que los nodos reciben la suma de mensajes del superpaso anterior, calculan un nuevo valor para la propiedad del vértice y posteriormente envían los mensajes a los nodos vecinos en el siguiente superpaso. A diferencia de la variante original de Pregel, el cálculo de los mensajes tiene acceso a las propiedades de los nodos de origen y destino y el cálculo de los mensajes se realiza en paralelo.

La teoría de grafos se ha vuelto más popular actualmente, esto es debido al poder de la computación disponible a mucho menos coste y a la llegada de nuevos marcos de procesamiento.

Un grafo es un estudio de las relaciones y está formado por nodos que están conectados por arcos o líneas.

A los arcos se les demonina múltiples en el caso en el que haya más de una línea entre los mismos nodos. También se puede dar la situación en la que un arco conecte a un nodo consigo

mismo, este caso se conoce como bucles propios o aristas propias. Cuando un grafo no tiene ni aristas propias ni arcos múltiples se denomina grafo simple. Un grafo con arcos múltiples es conocido como multigrafo.

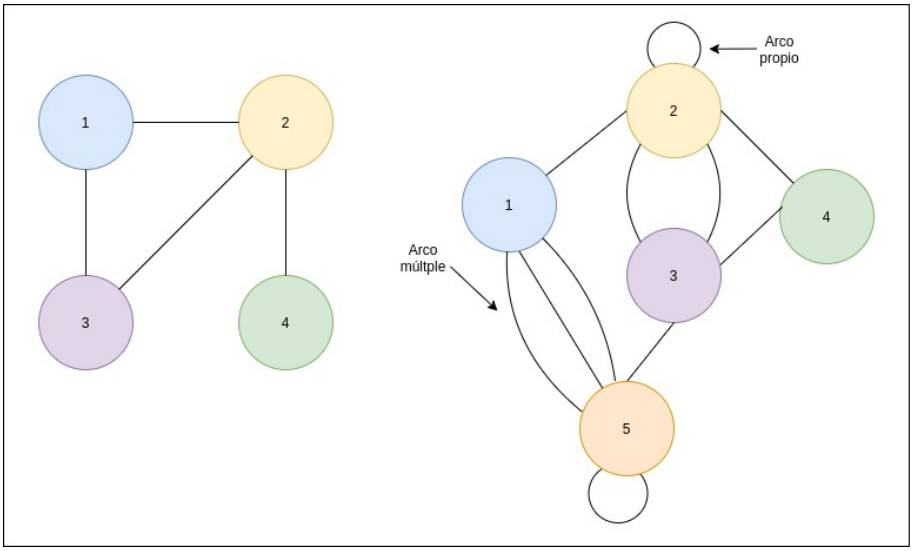


Figura 2.2: Ejemplo de grafo simple y multigrafo.

Un grafo puede ser dirigido, que implica que los arcos tienen una dirección asociada a ellos, o no dirigido. Un ejemplo de grafo dirigido puede ser la red de aguas de una ciudad debido a que cada tubería solo admite un único sentido para el agua. Sin embargo, la red de carreteras de un país sería ejemplo de grafo no dirigido ya que una carretera puede ser recorrida en varios sentidos.

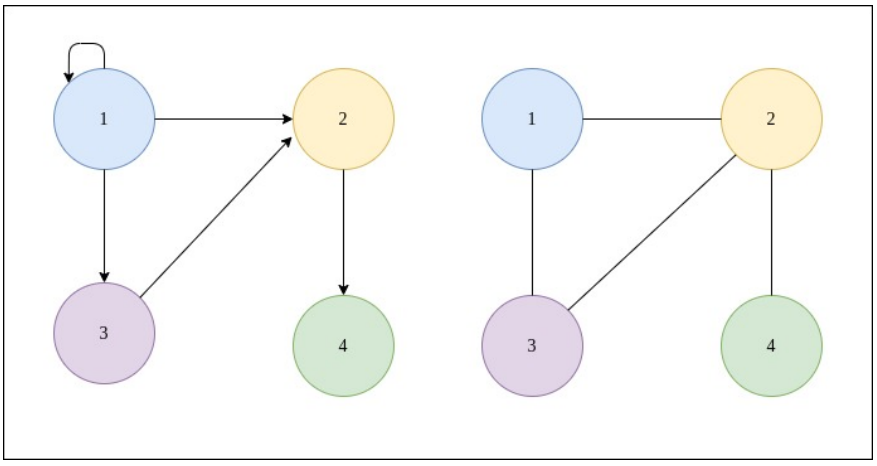


Figura 2.3: Ejemplo de grafo dirigido y no dirigido.

Un ciclo en un grafo dirigido es un bucle cerrado de arcos en el que las flechas de cada uno de ellos apuntan en la misma dirección alrededor del bucle. Algunas redes dirigidas tienen muchos ciclos de este tipo. Sin embargo, otras redes no tienen ciclos y se consideran acíclicas.

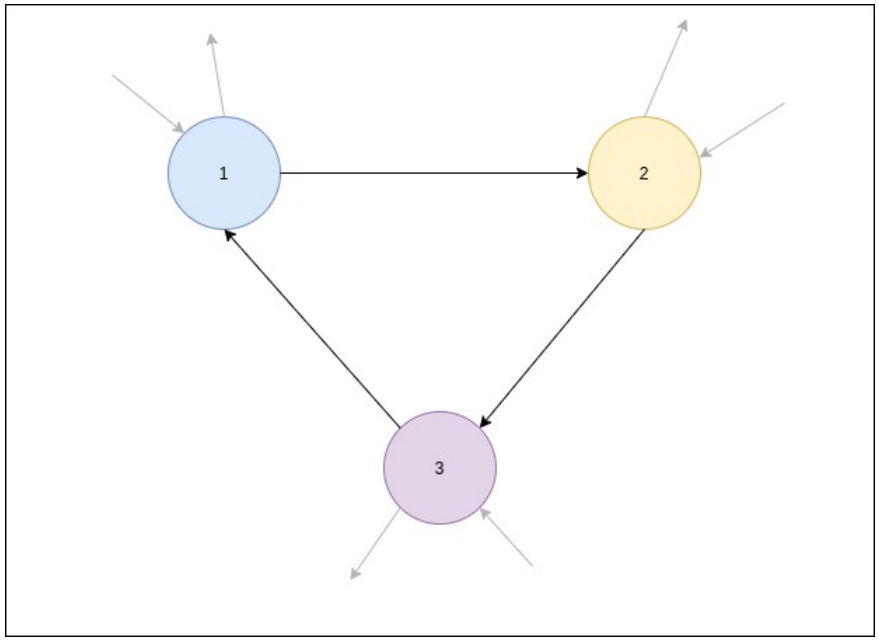


Figura 2.4: Ejemplo de grafo cíclico.

Un ejemplo típico de grafo cíclico es una red de citas de artículos, en el cual cuando se escribe un artículo solo se puede citar a otro en el caso en el que este ya haya sido escrito previamente. Lo que conlleva que todos los arcos apunten hacía atrás en el tiempo.

Otro tipo de grafo usado frecuentemente es el grafo bipartito. Este grafo representa redes con dos tipos distintos de nodos. Los arcos de este tipo de redes solo conectan nodos de tipos diferentes. Un ejemplo de este tipo de grafo podría ser una red de actores en la que un tipo de nodo serían los actores y el otro tipo representaría las películas. Los arcos unirían a los actores con las películas en las que ha participado. No es posible conectar actores con otros ni películas con otras debido a que son del mismo tipo.

El árbol es una de las tipologías de grafo más usadas para representar modelos básicos de red, ya que al ser un modelo directo, no dirigido y que no contiene bucles consigue que los cálculos básicos que se pueden realizar sobre estas sean especialmente sencillos de llevar a cabo. La propiedad más importante de esta tipología consiste en la certeza de que siemppre va a existir

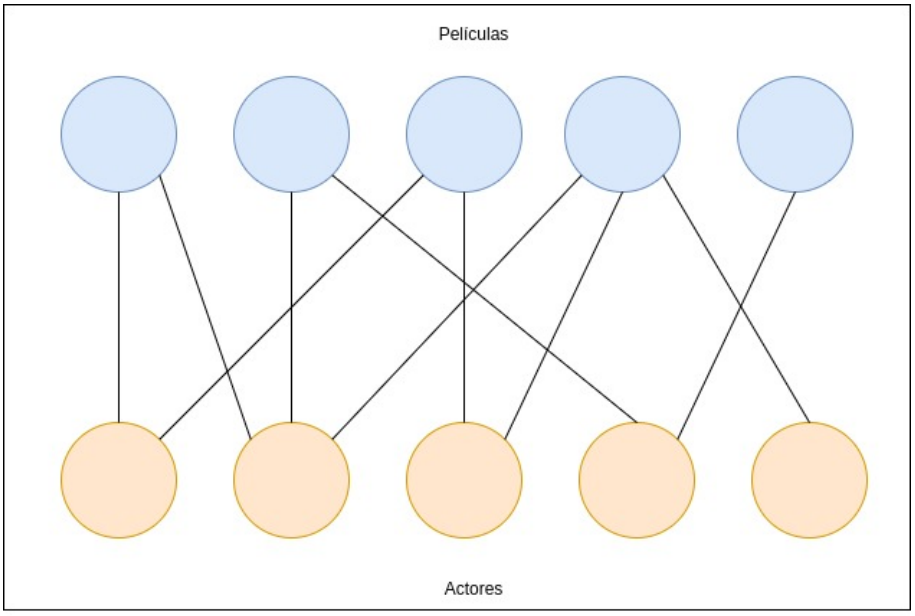


Figura 2.5: Ejemplo de grafo bipartito con actores y películas.

un único camino entre cualquier par de nodos. Otra de las propiedades a tener en cuenta es el conocimiento de que siempre va a tener $n$-1 arcos, siendo $n$ el número de nodos del grafo. La topología de red en árbol es ideal cuando las estaciones de trabajo están ubicadas en grupos, y cada grupo ocupa una región física relativamente pequeña. Un ejemplo de uso podría ser una configuración de red de área amplia (WAN).

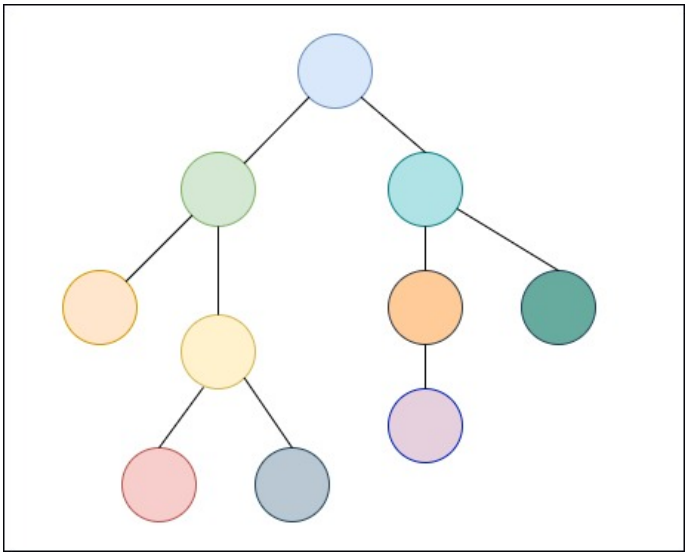

Figura 2.6: Ejemplo de grafo árbol.

La representación matemática fundamental de los grafos es la matriz de adyacencia. Esta puede ser utilizada para representar relaciones binarias, por lo que es una de las formas de denotar redes sociales en el análisis de redes. Tomamos como ejemplo el grafo simple representado en la figura 2.2, el cual es un modelo simple no dirigido con n nodos que han sido etiquetados de manera única de la forma 1..*n*. Esta red se define con una matriz *n* x *n*, siendo *n* el número de arcos, y con elementos $A_{ij}$ tales que

$$A_{ij} = \begin{cases} 1, & \text{si hay un arco entre nodos i y j} \\ 0, & \text{en el caso contrario} \end{cases} \tag{2.1}$$

La matriz de adyacencia de dicho grafo sería

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{2.2}$$

Podemos observar que los elementos de la diagonal de la matriz de adyacencia de un grafo simple, sin arcos propios, siempre van a ser cero. Otra de las propiedades a tener en cuenta es que la matriz es simétrica ya que si existe un arco entre $i$ y $j$, necesariamente habrá otro entre $j$ e $i$.

También es posible representar mediante una matriz de adyacencia arcos múltiples y arcos propios. Por ejemplo, la matriz de adyacencia del modelo multigrafo de la figura 2.2 sería

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 3 \\ 1 & 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 2 \end{pmatrix}. \tag{2.3}$$

Las aristas dobles entre nodos se representan como

$$A_{ij} = A_{ij} = 2 \ ,$$

mientras que la definición aceptada para los arcos propios es

$$A_{ii} = 2 \ ,$$

ya que se considera que el arco que va desde el nodo i a él mismo tiene dos extremos.

Uno de los conceptos de red más utilizados y más útiles es el de grado de un nodo. El grado en una red no dirigida se define como el número de aristas conectadas a él. En un grafo simple el grado de un nodo es igual a la cantidad de vecinos que tiene, pero es importante tener claro que este concepto se refiere a las aristas y no a los vecinos del nodo. Esto puede causar problemas en un multigrafo, ya que si un nodo tiene dos arcos paralelos con el mismo vecino, ambos constribuyen al grado. Este concepto puede escribirse en términos de matriz de adyacencia como

$$k_i = \sum_{j=1}^{n} A_{ij} \ . \tag{2.4}$$

Como se ha expuesto anteriormente, cada enlace en un grafo no dirigido tiene dos extremos, por tanto, para una cantidad $m$ de arcos habrá un número $2m$ de extremos de enlaces. Sabiendo que el número de extremos de arcos es igual a la suma de los grados de todos los nodos se obtiene:

$$2m = \sum_{i=1}^{n} k_i = \sum_{ij} A_{ij} \ . \tag{2.5}$$

El grado medio de un nodo en una red no dirigida es:

$$c = \frac{1}{n} \sum_{i=1}^{n} k_i \ . \tag{2.6}$$

Combinando esto con la ecuación 2.5 se obtiene:

$$c = \frac{2m}{n} . \tag{2.7}$$

Un concepto muy importante en el análisis de redes sociales es el de centralidad. Su utilidad proviene de la capacidad para ofrecer información sobre qué nodos son los más importantes de una red. Como la importancia de los nodos depende de muchos factores hay muchas medidas de centralidad para las redes. A continuación se resumen algunas de ellas.

- **Centralidad de grado.** Se puede considerar la medida de centralidad más sencilla, ya que se corresponde con el grado de un nodo, es decir el número de aristas conectadas a él. En las redes dirigidas, los nodos cuentan tanto con grado de entrada como de salida, y ambos puedes ser útiles como medidad de centralidad en diferentes análisis. Dependiendo de la naturaleza de la red, podemos interpretar la centralidad de grado de varias formas. Por ejemplo, en una red social, si el vínculo es de amistad, la centralidad de grado será la cantidad de amigos que tiene un nodo. Sin embargo, si el vínculo es de confianza, la medida pasará a ser el número de personas sobre las que el nodo que estamos analizando está en posición de influir.

- **Centralidad de vector propio.** Aunque la centralidad de grado pueda ser muy útil por su sencillez, no es la medida más efectiva, ya que no todos los nodos vecinos son necesariamente equivalentes. Por lo general, la importancia de un nodo en una red social se incrementa si cuenta con enlaces con otros nodos que también son importantes e influyentes. Por tanto, la centralidad no sólo tiene que ver con el número de personas sino también con quiénes son. En lugar de otorgar un punto por cada vecino de la red que tenga un nodo, la centralidad de vector propio añade un número de puntos en relación a la puntuación de centralidad de los vecinos.

  En una red no dirigida de $n$ nodos, la centralidad del vector propio $x_i$ del nodo $i$ se define como proporcional a la suma de centralidades de sus vecinos, de modo que usando la matrix de adyacencia se tiene:

$$x_i = k^{-1} \sum_{j=1}^{n} A_{ij} x_j, \tag{2.8}$$

donde $k^{-1}$ es una constante de proporcionalidad llamada *valor propio*.

Con la centralidad de vector propio definida de esta manera, un nodo puede conseguir una centralidad elevada teniendo muchos vecinos con una centralidad modesta, o teniendo pocos vecinos pero con una centralidad alta, o ambas cosas. Esto consta de lógica ya que se puede ser influyente conociendo a mucha gente o conociendo a unos pocos muy influyentes.

- **Centralidad por cercanía.** La centralidad de cercanía es una puntuación que mide la distancia media de un nodo a otros nodos. Para esta medida se utiliza el concepto del camino más corto de un grafo. Este camino, también conocido como geodésico, es el más corto entre un par de nodos dado, es decir, el que atraviesa el menor número de aristas. La distancia más corta o distancia geodésica entre dos nodos, es la longitud del camino más corto en términos de números de aristas. Una vez se conoce este concepto, siendo $d_{ij}$ la distancia más corta desde el nodo $i$ al nodo $j$, entonces la distancia media más corta desde $i$ a cualquier nodo de la red es:

$$l_i = \frac{1}{n} \sum_j d_{ij} \ . \tag{2.9}$$

La cercanía es una medida inversa de la centralidad, en el sentido de que los números grandes indican que un nodo es muy periférico, mientras que las cantidades pequeñas muestran que un nodo es más central. En una red social una persona con una distancia media más baja con respecto a los demás podría descubrir que sus opiniones se extienden por la comunidad más rápidamente que las de otras personas. Por lo tanto, en redes sociales, los investigadores suelen calcular la inversa de $l_i$. Esta inversa se denomina centralidad de cercanía $C_i$:

$$C_i = \frac{1}{l_i} = \frac{n}{\sum_j d_{ij}} \ . \tag{2.10}$$

- **Centralidad de intermediación.** La centralidad de intermediación mide el grado en que un nodo se encuentra en los caminos entre otros nodos. Tomando un nodo focal, se calcula para

cada par de nodos distinto al elegido, qué proporción de todos los caminos más cortos de uno a otro pasan por el nodo focal. La fórmula de la intercentralidad del nodo j viene dada por

$$b_j = \sum_{i<k} \frac{g_{ijk}}{g_{ik}} , \tag{2.11}$$

donde $g_{ijk}$ es el número de caminos geodésicos que conectan $i$ y $k$ a través de $j$, y $g_{ik}$ es el número total de trayetorias geodésicas que conectan $i$ y $k$.

Los nodos con alta centralidad por intermediación pueden tener más influencia dentro de una red, por su control sobre la información que pasa entre ellos.

# Capítulo 3

# Datasets

## 3.1. Dataset pequeño: Stack Overflow

Como primer dataset para poner en práctica lo aprendido de las tecnologías comentadas anteriormente se ha elegido uno de pequeñas dimensiones con datos extraídos de Stack Overflow.

Stack Overflow es una web privada en la que se encuentran más de 20 millones de preguntas y respuestas sobre diferentes temáticas de programación e informática. La plataforma fue fundada en 2008 y goza de gran popularidad.

El contenido del dataset está organizado en dos tablas:

- **stacknetworknodes**

  Esta tabla contiene los nodos de la red y está formada por las siguientes columnas:

  - **name**. Nombre del nodo.
  - **group**. Grupo al que pertenece el nodo.
  - **nodesize**. Tamaño de nodo basado en la frecuencia de uso de esa etiqueta tecnológica.

- **stacknetworklinks**

  Esta tabla contiene enlaces de la red y está formada por las siguientes columnas:

  - **source**. Etiqueta tecnológica de origen.

- **target**. Etiqueta tecnológica de destino.
- **value**. valor del enlace entre cada par.

## 3.2. Dataset grande: IMDb

Como segundo dataset se ha elegido uno de mayores dimensiones que incluye una extensa cantidad de datos extraídos de IMDb.

IMDb es la fuente más popular y autorizada del mundo para el contenido de películas, televisión y celebridades, diseñada para ayudar a los fans a explorar el mundo de las películas y programas y decidir qué ver. Fue fundada en 1990 y, posteriormente, se convirtió en filial de Amazon en 1998. La plataforma incluye millones de películas, programas de televisión y de entretenimiento, así como miembros del reparto y del equipo.

Este dataset está formado por 4 tablas, pero nos centraremos en la correspondiente a las películas:

- **IMDb movies**

  Esta tabla contiene 85.855 películas y está compuesta por 22 columnas. Los atributos más destacables se encuentran en las siguientes columnas:

  - **imdb title id**. Id del título de la película.
  - **title**. Título de la película.
  - **year**. Año de estreno de la película.
  - **date of release**. Fecha de estreno de la película.
  - **movie genre**. Géneros a los que pertenece la película.
  - **duration**. Duración de la película.
  - **movie country**. País o países de los que es originaria la película.
  - **director**. Director o directores de la película.
  - **actors**. Actores que participan en la película.
  - **description**. Sinopsis de la película.
  - **avg vote**. Puntuación media de todos los votos recibidos.

- **reviews from users**. Cantidad de comentarios de usuarios que tiene la película.
- **reviews from critics**. Cantidad de comentarios de críticos que tiene la película.

# Capítulo 4

# Experimentos

## 4.1. Experimento Stack Overflow

Para comenzar los experimentos es necesario construir una *SparkSession*. Esta es una instancia que aporta el punto de entrada a toda la funcionalidad de Spark.

```
spark = (SparkSession
        .builder
        .appName("Stackoverflow")
        .getOrCreate())
```

Figura 4.1: SparkSession builder

Con el dataset de datos extraído de Stack Overflow que se ha comentado con anterioridad se ha decidido realizar el siguiente experimento:

Dado que se cuenta con lenguajes de programación como nodos, los cuales están organizados por grupos, y los enlaces entre ellos vienen dados por las etiquetas de origen y destino, se va a analizar la relación entre los diferentes grupos en función de las acciones de los usuarios de Stack Overflow.

Para ello se procede a generar un grafo con GraphFrame y, para una mejor visualización del mismo, se decide representarlo por medio de NetworkX, una biblioteca de Python para el estudio

de grafos y análisis de redes, y siguendo el *Fruchterman-Reingold force-directed algorithm*, que consiste dando como resultado el grafo mostrado a continuación en la figura 4.2.

En el grafo podemos observar con bastante claridad cómo se organizan los diversos grupos y las relaciones existentes entre las demás agrupaciones.

En el centro nos encontramos con uno de los grupos con más importancia, el cual está formado por lenguajes como JavaScript, PHP, HTML, jQuery, CSS, MySQL, AngularJS, entre otros.

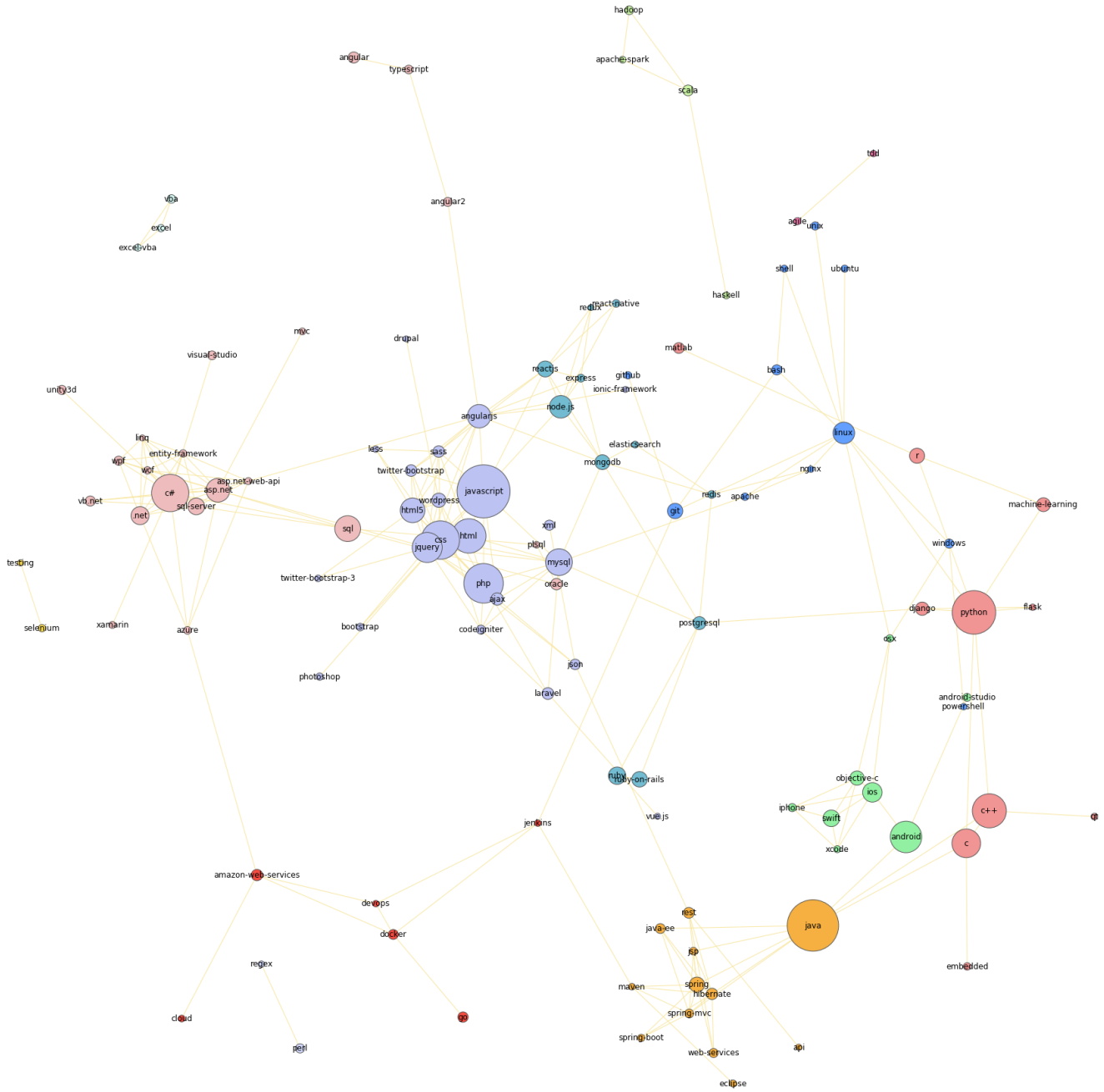


Figura 4.2: Grafo StackOverflow.

# Capítulo 5

# Conclusiones

## 5.1. Resumen de resultados

```
aspell --lang=es_ES -c memoria.tex
```

## 5.2. Estimación de esfuerzo

1. a

2. b

## 5.3. Asignaturas relacionadas

## 5.4. Lecciones aprendidas

1. Aquí viene uno.

2. Aquí viene oto.

## 5.5. Trabajos futuros

# Apéndice A

# Manual de usuario

# Apéndice B

# Bibliografía

https://hiramnoriega.com/20652/stack-overflow-que-es-caracteristicas/

# Bibliografía

[1] M. Newman. *Networks*. Oxford university press, 2018.

[2] S. Wasserman, K. Faust, et al. *Social Network Analysis: Methods and Applications*. Cambridge university press, 1994.