



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG Ingeniería Informática:
NetExtractor 3.0



Presentado por David Hernando Martín
en Burgos el 12 de junio de 2023
Tutores D. José Manuel Galán Ordax
y Dña. Virginia Ahedo García

D. José Manuel Galán Ordax y Dña. Virginia Ahedo García,
profesores del departamento de Ingeniería Civil, área de Organización de
Empresas.

Exponen:

Que el alumno D. David Hernando Martín, con DNI 71482276R, ha
realizado el Trabajo final de Grado en Ingeniería Informática titulado
NetExtractor 3.0.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección
del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 12 de Junio de 2023

Vo. Bo. del Tutor:

Vo. Bo. del Tutor:

Resumen

En la actualidad, el estudio de quién conecta con nosotros y con quienes estamos conectados, marca nuestro día a día sin apenas darnos cuenta. Términos como “*amigos*” en Facebook, “*seguidores*” en Twitter o directamente la terminología empleada en LinkedIn son claros ejemplos de ello. Quién no ha recibido la pregunta en los últimos años ¿Contactos en los últimos días?

La ciencia de redes es la encargada de estudiar y analizar todas estas relaciones que tanta importancia tienen. Permite a los analistas obtener información muy valiosa de cara a futuros avances y estudios. Sin embargo, las herramientas para el estudio de estos datos no son especialmente vistosas ni sencillas de manejar, lo cual le resta atractivo para el usuario que no esté familiarizado con esta disciplina.

Por ello surgió el proyecto de NetExtractor 3.0, para continuar con el trabajo realizado por otros compañeros a la hora de realizar una aplicación que acerque la ciencia de redes a todos los públicos.

Para ello, de una forma sencilla e interactiva, la aplicación permite una visualización estática y dinámica de las redes de interacción de novelas, películas y obras de teatro, que el usuario pueda elegir en base a sus intereses.

Descriptores

Generador de redes de interacción, evolución de red dinámica, Python, aplicación web, obras de teatro.

Abstract

Nowadays, the study of who connects with us and with whom we are connected marks our day to day without hardly realizing it. Terms such as "friends" on Facebook, "followers" on Twitter or directly the terminology used on LinkedIn are easy examples of this. Who has not received the question in recent years, Contacts in recent days?

Network science oversees studying and analyzing all these relationships that are so important. It allows analysts to obtain very valuable information for future advances and studies. However, the tools for studying these data are not particularly attractive or easy to use, which makes them less attractive for users who are not familiar with this discipline.

That is why the NetExtractor 3.0 project arose, to continue with the work carried out by other colleagues when creating an application that brings network science closer to all audiences.

To do this, in a simple and interactive way, the application allows a static and dynamic visualization of the interaction networks of novels, films and plays, which the user can choose based on their interests.

Keywords

Interaction network generator, dynamic network evolution, Python, web application, plays

ÍNDICE GENERAL

ÍNDICE GENERAL	I
ÍNDICE DE FIGURAS	IV
A. INTRODUCCIÓN.....	1
A.1 Estructura de la memoria	2
A.2 Enlaces adicionales	3
B. OBJETIVOS DEL PROYECTO.....	4
B.1 Objetivos generales.....	4
B.2 Objetivos técnicos.....	5
B.3 Objetivos personales	6
C. CONCEPTOS TEÓRICOS	7
C.1 EPUB	7
C.2 Redes y redes complejas	8
C.3 Grado de los nodos.....	10
C.4 Medidas de distancia.....	12
C.5 Medidas de clustering	13
C.6 Medidas de centralidad, Centralidad y centralización	15
C.7 Grupos y comunidades.....	21
C.8 Detección de roles.....	27
C.9 Scraping Web.....	29
C.10 Redes Dinámicas	30
C.11 Obras de teatro	30
C.12 Corpus Lingüístico	31
D. TÉCNICAS Y HERRAMIENTAS	32
D.1 Metodología ágil – Scrum.....	32
D.2 Herramienta de control de versiones.....	32
D.3 Herramienta de gestión del proyecto	33

D.4	Herramienta para realización de la documentación	33
D.5	Herramienta para gestión y edición del repositorio local-remoto.....	33
D.6	Herramienta para la gestión de referencias bibliográficas	34
D.7	Lenguaje de programación.....	34
D.8	Interfaz gráfica	34
D.9	Herramienta para la interfaz.....	35
D.10	Programación en el lado del cliente	35
D.11	Tablas con filtro	36
D.12	<i>Scraping</i> Web.....	36
D.13	Herramienta para la obtención de datos de las obras de teatro	36
D.14	Funcionalidad etnia y sexo	36
D.15	Generación de la red estática.....	37
D.16	Generación de la red dinámica.	37
D.17	Visualización de la red	37
D.18	Herramienta para la exportación a Gexf Dinámica	38
D.19	Herramientas para la exportación a vídeo	38
D.20	Realización de la wiki	39
D.21	Herramienta para gestionar las traducciones.....	39
D.22	Herramienta para generación de diagramas	39
D.23	Herramienta para la obtención de gráficos de sprint.....	40
D.24	Herramienta para albergar la aplicación.....	40
E.	ASPECTOS RELEVANTES DE DESARROLLO DEL PROYECTO...	41
E.1	Inicio del proyecto	41
E.2	Metodologías.....	41
E.3	Formación	42
	Dracor API, NetworkX y Netwulf.....	42
	Datatables.....	43

E.4	Desarrollo de nuevas implementaciones.....	44
	Módulo Obras de teatro	44
	Redes para obras de teatro, función Read_Gexf.....	44
	Mejoras gráficas y visuales.....	45
	Nuevas implementaciones dinámicas	46
E.5	Corrección de errores.....	47
	Errores por librerías desactualizadas	47
	Errores propios y refactorización.....	48
	Errores de versiones previas	49
E.6	Internacionalización.....	49
E.7	Ayuda al usuario	49
E.8	Wiki.....	50
E.9	Despliegue.....	50
F.	TRABAJOS RELACIONADOS	51
F.1	UbuBookNet	51
F.2	NetExtractor 1.0 y 2.0.....	52
F.3	Network of Thrones	52
F.4	Evaluating named entity recognition tools for extracting social networks from novels	53
F.5	Character network analysis of Émile Zola's Les Rougon-Macquart....	54
G.	CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS	55
G.1	Conclusiones	55
G.2	Líneas de trabajo futuras	55
	BIBLIOGRAFÍA	57

ÍNDICE DE FIGURAS

Ilustración 1. Red Compleja	9
Ilustración 2. Centralidad de Grado	16
Ilustración 3. Centralidad de Cercanía.....	18
Ilustración 4. Centralidad de Intermediación.....	19
Ilustración 5. Centralidad de vector propio	20
Ilustración 6. PageRank	21
Ilustración 7. Gráfico de detección de roles	28

A.INTRODUCCIÓN

La ciencia de redes es una disciplina académica que se ocupa de investigar las redes complejas y sus características principales. En este campo, los elementos o actores se representan mediante nodos (o vértices), mientras que las conexiones entre ellos se representan como enlaces (o aristas). Su estudio emergió en diversas disciplinas como un medio para analizar datos relacionales complejos y por ello se utilizan teorías y métodos provenientes de diversas áreas, como la teoría de grafos en matemáticas, la mecánica estadística en física, la minería de datos y la visualización de información en ciencias de la computación, la inferencia estadística en estadística y la estructura social en sociología.[1]

En la actualidad, la ciencia de redes tiene un gran impacto a nivel social y científico en diversos aspectos. Empresas exitosas como Google, Facebook y Amazon basan su tecnología y modelo de negocio en las redes. Por ejemplo, el algoritmo de búsqueda de Google, conocido como PageRank, utiliza redes para dirigir las búsquedas hacia las páginas más importantes.

En el campo de la salud, la importancia de las redes moleculares ha llevado al surgimiento de la biología de redes, que busca comprender el comportamiento de las redes celulares. La medicina de redes, por su parte, busca descubrir el papel de las redes en las enfermedades humanas y es especialmente relevante en el desarrollo de fármacos. La farmacología en red tiene como objetivo final desarrollar medicamentos que puedan curar enfermedades sin efectos secundarios significativos.

El surgimiento de un marco basado en redes ha traído un cambio fundamental al ofrecer un nuevo nivel de predictibilidad para los modelos de epidemias. Hoy en día, la predicción de epidemias es una de las aplicaciones más activas de la ciencia de redes y se ha demostrado su utilidad para evitar desastres aún mayores.[2]

Estos son solo alguno de los ejemplos más importantes donde puede aplicar, pero ya podemos observar que tiene una gran importancia. Sin embargo, las herramientas empleadas hoy en día para el desarrollo y estudio de las redes, no facilitan el uso al usuario no especializado, dificultando su manejo y el interés, provocando un mayor desconocimiento y una menor captación de posibles interesados en esta área.

Es por todo esto, que el enfoque principal de este proyecto, del cual ya se han desarrollado hasta tres versiones, sea acercar el manejo de las redes a usuarios que carezcan de conocimientos previos, mediante una aplicación web con una interfaz sencilla y fácil de manejar. Si bien es cierto que una interfaz con la que es fácil interactuar ayuda mucho en la curva de aprendizaje y captación del usuario, hace falta algo más que un buen diseño para llamar la atención y aquí entra el factor de los gustos personales. NetExtractor 3.0 incorpora el análisis y visualización de redes de interacción de obras de teatro a los módulos ya existentes de películas y novelas, permitiendo al usuario seleccionar entre multitud de opciones en base a sus gustos. También se ha intentado realizar mejoras de diseño de cara a facilitar el manejo al usuario y la incorporación de nuevas funcionalidades a las ya implementadas anteriormente en versiones previas.

A.1 Estructura de la memoria

La memoria va a tener la siguiente estructura:

- **Introducción:** Descripción inicial del proyecto, estructura y enlaces a los recursos empleados durante su desarrollo.
- **Objetivos del proyecto:** Descripción de los objetivos que se quieren alcanzar con la realización del proyecto: generales, técnicos y objetivos personales.
- **Conceptos teóricos:** Descripción de aquellos conceptos considerados fundamentales para entender el proyecto.
- **Técnicas y herramientas:** Listado de las herramientas y técnicas empleadas durante el desarrollo del proyecto
- **Aspectos relevantes del desarrollo:** Aspectos de alta relevancia ocurridos durante la realización del proyecto: contratiempos, problemas y soluciones aplicadas y cambios de decisión
- **Trabajos relacionados:** Trabajos previos sobre el tema que trata el proyecto o similares que sirven como referencia.
- **Conclusiones y líneas de trabajo futuras:** Conclusiones sacadas una vez finalizado el proyecto y posibles mejoras/implementaciones que permitirían mejorar el proyecto actual (funcionalidades adicionales, optimización, etc.).

A su vez también se proporciona un fichero con los anexos que contienen:

- **Planificación del proyecto:** Planificación temporal y estudio económico.
- **Especificación de requisitos:** Especificación de los requisitos según los objetivos generales marcados para el proyecto.
- **Especificación de diseño:** Diseño de los datos, librerías y paquetes empleados e interfaz de la aplicación.
- **Manual del programador:** Pequeño manual para desarrolladores con la estructura del proyecto, instalación de librerías para el modo local, cómo se ejecuta la aplicación y la realización de las pruebas.
- **Manual de usuario:** Se establece una guía o manual de usuario para su correcto entendimiento y poder manejar la aplicación sin mayor problema.

A.2 Enlaces adicionales

Los enlaces adicionales del proyecto que podemos encontrar son:

- **Página web de la aplicación:**
 - <https://netextractor3.up.railway.app/>
- **Wiki de la aplicación:**
 - <http://wikinetextractor-3-0.wikidot.com/guia-de-usuario>
- **Repositorio:**
 - <https://github.com/dhm1002/NetExtractor-3.0>
- **Youtube:**
 - Ejecución simple: <https://youtu.be/F1K75OnCqls>
 - Ejecución detallada: https://youtu.be/I_fl5TdMJM0

B.OBJETIVOS DEL PROYECTO

En este apartado se van a definir los objetivos del proyecto. Primero los objetivos de carácter general, aquellos que han sido marcados al comienzo del proyecto como implementaciones o mejoras a las versiones previas. Después los objetivos técnicos, acerca del uso de herramientas y metodologías que vamos a emplear y poner en práctica en un proyecto real. Por último, mis objetivos personales de cara al desarrollo de este proyecto, en que me quiero enfocar y que quiero haber aprendido una vez concluya el trabajo.

Este proyecto parte de varios trabajos previos y es la versión 3.0 del proyecto NetExtractor. La primera versión fue desarrollada por el alumno Jorge Navarro González y partía de un trabajo previo titulado Ububooknet, de Luis Miguel Cabrejas Arce. Esta versión realizaba la extracción de las interacciones de personajes en novelas o películas en función del formato. Una vez obtenidas las interacciones elaboradas la red y permitía al usuario su visualización junto con la posibilidad de elaborar un informe.

La última versión (2.0) fue presentada por el alumno Alberto Díez Busto el 7 de Julio de 2022. Esta última versión, añadía la visualización y la obtención de los informes para las redes dinámicas, permitiendo ver la evolución de la red y obtener sus correspondientes métricas.

B.1 Objetivos generales

En este apartado se van a describir los objetivos generales del proyecto, es decir, los objetivos de carácter general, aquellos que han sido marcados al comienzo del proyecto como implementaciones o mejoras a las versiones previas

- Ampliar la aplicación web para permitir introducir obras de teatro, a través de la API de Dracor <https://dracor.org/> manteniendo las funcionalidades previas.
- Añadir opciones para la implementación dinámica, como la visualización automática o la exportación dinámica a Gexf.
- Mejorar la interfaz general de la aplicación de forma que sea más vistosa y atractiva al usuario.

- Actualización de la wiki a modo de guía de usuario puesto que la que hay actualmente hace referencia al proyecto 1.0. Dispondrá del manual de usuario definido en el anexo para resolver cualquier duda que le surja al usuario.
- Solucionar errores que hayan podido aparecer con la actualización de librerías, para mantener las funcionalidades de las versiones anteriores.
- Encontrar una opción para el despliegue dado que las implementaciones previas empleaban Heroku y ahora ha pasado a ser de pago.

B.2 Objetivos técnicos

En este apartado se definirán los objetivos de carácter técnico, es decir acerca del uso de herramientas y metodologías que vamos a emplear y poner en práctica en un proyecto real

- El desarrollo del proyecto se realizará conforme a la metodología Scrum [3] estudiada durante la carrera para el desarrollo de software.
- Utilizar ZenHub como herramienta para la gestión de proyectos.
- Utilizar GitHub como herramienta de control de versiones.
- Utilizar Visual Studio Code para gestionar el repositorio local y remoto.
- Utilizar Python como lenguaje de programación base. Librerías de Python a utilizar serán: Networkx, NumPy, Matplotlib, Python-louvain, Scipy, BeautifulSoup4, Ply, Flask_Babel, gexfpy, DyNetX, request.
- Utilizar Flask para el desarrollo web.
- Utilizar el patrón de diseño Modelo-Vista-Presentador
- Utilizar Wikidot para la realización de la wiki.
- Hacer disponible la aplicación y la wiki a través de la web.

B.3 Objetivos personales

En este apartado se van a definir una serie de objetivos personales marcados:

- Mejorar mis habilidades en lenguajes de diseño web como pueden ser CSS y HTML.
- Profundizar en el trabajo con la ciencia de redes, puesto que es una disciplina que me llama la atención, de cara futuros estudios superiores.
- Poner en práctica todos los conocimientos adquiridos sobre Ciencia de Redes.
- Mejorar mi organización de trabajo a través de la metodología SCRUM, la cual ya había empleado en mi trabajo, con muy buenos resultados.
- Profundizar en el desarrollo web con entornos Python y en lo necesario para la gestión de sesiones y despliegues, en vista a proyectos futuros.

C.CONCEPTOS TEÓRICOS

Para todos los conceptos teóricos relacionados con grafos se han usado como referencias principales los siguientes libros que deben ser mencionados previamente:

- Network Science Book [2] de Albert-László Barabási
- Networks: An Introduction [4] de Mark Newman
- Introduction to Graph Theory [5] de Richard J Trudeau
- Networks, Crowds, and Markets [6] de David Easley y Jon Kleinberg
- Social and Economic Networks [7] de Matthew O. Jackson

C.1 EPUB

Un fichero EPUB (*Electronic Publication*) es un formato de archivo utilizado para libros electrónicos y publicaciones digitales. Se trata de un estándar abierto muy utilizado que permite la distribución y visualización de contenido en diversos dispositivos electrónicos, como lectores de libros electrónicos, tabletas y teléfonos inteligentes.[8]

El formato EPUB se basa en estándares web abiertos, como pueden ser HTML y CSS, lo que facilita la creación de libros electrónicos con diseño y estructura flexibles. Los ficheros EPUB son archivos comprimidos que contienen varios de estos componentes, incluyendo archivos HTML, CSS, imágenes y metadatos, los cuales se combinan para formar un libro electrónico completo.

Uno de los ficheros más importantes dentro de la estructura de un EPUB, es el llamado “container.xml”. Este fichero es el que va a dirigir el sistema de lectura a la raíz de la publicación, indicando el contenido de todos los documentos que se incluyen en la publicación y sus respectivos recursos. En este fichero existe una etiqueta llamada “*spine*”, la cual va a ser la encargada de indicar el orden en el que se van a ir sucediendo los distintos archivos a la hora de leer el EPUB.

Una de las ventajas principales del formato EPUB es su diseño responsive, el cual permite que el contenido se adapte automáticamente al tamaño de pantalla del dispositivo en el que se visualiza, lo que permite una lectura cómoda y fluida. Además, los ficheros EPUB suelen admitir características como marcadores, búsqueda de texto, ajuste del tamaño de fuente y notas al pie, entre otras.

C.2 Redes y redes complejas

Probablemente este sea uno de los apartados teóricos más importantes del proyecto, puesto que es el eje central de nuestro desarrollo. Vamos a comenzar definiendo lo que es una red, antes de dar paso a la explicación sobre las redes complejas, que son las que nosotros hemos empleado.

Una red hace referencia a un conjunto de nodos interconectados mediante enlaces que representan diferentes tipos de información y pueden tener distintas relevancias dentro de la red. El tamaño de una red depende del número de nodos que la componen, lo cual será relevante para las redes complejas como veremos a continuación.

En lo relativo a los nodos, una red puede tener nodos de una sola clase, conocidas como redes unimodales, como por ejemplo una red en la que los nodos representan personas, como pueden ser las redes sociales. También puede haber nodos de dos clases, denominadas redes bimodales, como en el caso en el que un nodo representa una red de empresas privadas y otro nodo representa organizaciones sin ánimo de lucro, sobre las que las primeras realizan donaciones. Por último, existen redes multimodales en las que los nodos pueden pertenecer a múltiples clases, como podría ser una red de actores, directores y guionistas. Como podemos observar, no significa que sean necesariamente elementos diferentes, sino que cumplan funciones o roles diferentes. Por eso los nodos también reciben el nombre de actores.

En base a los enlaces, se pueden distinguir dos tipos de redes: dirigidas y no dirigidas. Las redes dirigidas presentan una dirección y un sentido en la relación entre los nodos, como por ejemplo una red de citaciones de artículos científicos. En algunas ocasiones, puede haber enlaces dirigidos que tienen origen y destino en el mismo nodo y se denominan "auto-enlaces". Por otro lado, las redes no dirigidas solo indican la existencia de una relación entre los nodos, sin especificar un sentido particular, como puede ser el caso de una red que muestre las relaciones de pareja con otras personas.

Por último, en lo referente a las características simples de las redes, En función del número de enlaces entre pares de nodos, se pueden distinguir redes simples y múltiples. Las redes simples tienen como máximo un enlace entre cada par de nodos, mientras que las redes múltiples tienen más de un enlace entre los nodos. Además, las redes pueden clasificarse según cómo se representa el enlace, siendo una red binaria cuando simplemente representa la relación entre dos nodos sin dar más información, o

una red pesada cuando los enlaces tienen un peso asociado, como en el caso de la red de interacción entre actores y directores, en el caso de que se indicara cuantas veces han trabajado juntos.

Una vez hemos visto las características básicas que conforman las redes, podemos dar paso a las redes complejas, las cuales se definen básicamente por tener una estructura de red que exhibe características y propiedades no triviales, es decir, que no siguen necesariamente un patrón regular y se puede obtener información a partir de ellas. Estas redes son distintivas debido a su topología y comportamiento, que a menudo presentan una organización no aleatoria y propiedades emergentes.

Las Ilustración 1. Red Compleja se caracterizan por ciertas propiedades, como la conectividad no homogénea, la presencia de comunidades o clústeres, la distribución no uniforme de grados de conectividad de los nodos y la propiedad de pequeño mundo. Estas propiedades las explicaremos a continuación.

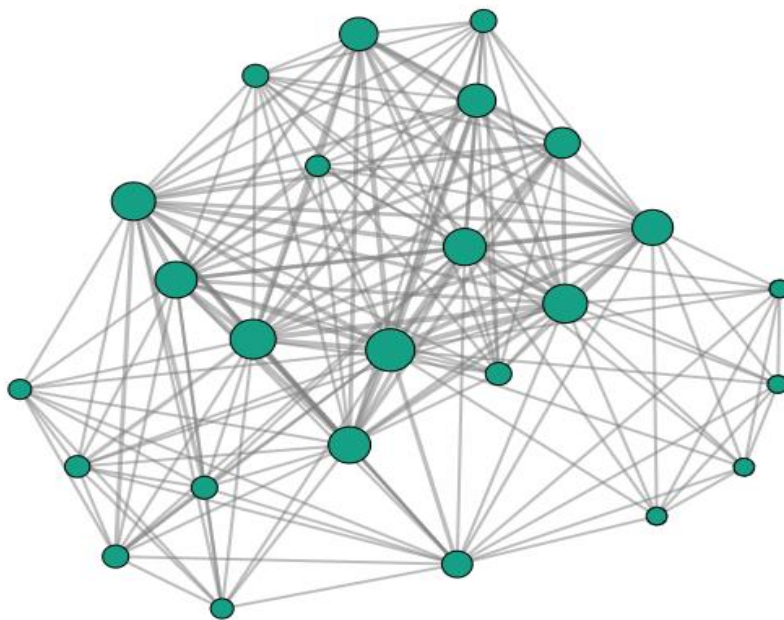


Ilustración 1. Red Compleja

Por último, hay que mencionar que, para nuestro proyecto, las redes que vamos a analizar son:

- No dirigidas, puesto que son bidireccionales. Solo contamos como enlace en el momento que hay una interacción, sin valorar quién la inicia.
- Simples, ya que entre dos nodos solo puede existir un enlace.

- Pesadas, al establecer un peso para los enlaces en función el número de veces que se establece una interacción entre personajes.

C.3 Grado de los nodos

El grado de un nodo en una red, se refiere al número de enlaces que tiene ese nodo con otros nodos de la propia red. Es una medida fundamental para analizar la conectividad y la importancia de un nodo dentro de una red.

En una red no dirigida, el grado de un nodo se calcula contando el número de enlaces que están conectados a ese nodo, mientras que en una red dirigida se debe realizar una distinción entre el grado de salida y el de entrada, en función de la dirección que aplique para cada enlace.

El grado es una medida importante para determinar la importancia o centralidad de un nodo dentro de una red. Los nodos con un grado alto suelen tener una mayor, mientras que los nodos con un grado bajo pueden tener una influencia limitada al disponer de una menor cantidad de enlaces.

Grado medio

Para realizar el cálculo del grado medio, dependerá de si la gráfica es dirigida o no:

- **Red no dirigida:** Como no se tiene en cuenta la salida o entrada de los enlaces, el grado medio (1) será el sumatorio de todos los grados dividido entre el total de nodos. Como ya sabemos, el grado es el número de enlaces. Al contabilizar cada enlace por los dos nodos que le unen tendremos que el grado medio final será dos veces el número de enlaces(L) dividido entre el número de nodos(N).

$$\langle k \rangle = \frac{L}{N}$$

1

- **Red dirigida:** En este caso, sí que se tiene en cuenta desde que nodo parte el enlace para definir grado de entrada y grado de salida. El grado medio total (2) será la suma del grado medio de entrada y el grado medio de salida, o lo que es lo mismo, el total de enlaces (L) multiplicado por dos entre el número de nodos (N).

$$\langle k \rangle = \frac{2L}{N}$$

2

Distribución de grado

La distribución de grado (3) de una red se define como la fracción de nodos en la red con grado k. Así si hay N nodos en total en una red y N de ellos tienen grado k, tenemos:

$$P(k) = N_k / N$$

3

La distribución de grado es muy importante para estudiar todo tipo de redes. El modelo de red más simple, por ejemplo, el gráfico aleatorio (*modelo Erdős-Rényi*), en el que cada uno de los N nodos tiene una probabilidad p de estar conectado de forma independiente, tiene una distribución binomial de grados k, lo que indica que no suele haber grandes diferencias entre los grados de los nodos.

Sin embargo, la mayoría de las redes en el mundo real, tienen distribuciones de grados muy diferentes a esta. La mayoría tienen un alto sesgo hacia la derecha, lo que significa que la gran mayoría de los nodos tienen un grado bajo, pero un pequeño número, conocido como "hubs" que veremos más adelante, tiene un grado alto. Se argumentó que algunas redes, en particular Internet, tienen distribuciones de grados que siguen aproximadamente una ley de potencia (4):

$$P(k) \sim k^{-\gamma}$$

4

Este tipo de redes se denominan redes libres de escala y han atraído una atención particular por sus propiedades estructurales y dinámicas.[9]

Densidad

La densidad (5) es una medida de red que indica qué tan conectados están los nodos entre sí. Se calcula comparando el número de conexiones enlaces existentes con el número total de enlaces posibles en la red.

En una red no dirigida, donde los enlaces son simétricos y bidireccionales, la densidad se calcula dividiendo el número de enlaces existentes entre los nodos por el número total de enlaces posibles, siendo la fórmula para calcular la densidad en una red:

$$D = \frac{2 * L}{N * (N - 1)}$$

5

La densidad resultante de esta fórmula es un valor entre 0 y 1. Un valor de densidad cercano a 1 indica que la red está altamente conectada, ya que la mayoría de los enlaces posibles existen. Por otro lado, un valor de densidad cercano a 0 indica que la red está poco conectada, ya que solo unos pocos enlaces posibles se han realizado.

Para una red dirigida, se utilizan fórmulas diferentes para calcular la densidad, considerando los enlaces de entrada y salida de cada nodo por separado.

C.4 Medidas de distancia

Dentro de todas las posibles medidas de distancia dentro de una red, la más importante es la distancia geodésica. Es la medida de longitud del camino más corto entre dos nodos en una red, representado la mínima cantidad de enlaces o saltos necesarios para ir desde un nodo hasta otro en una red. Es una medida importante en el análisis de redes, ya que proporciona información sobre la eficiencia y la accesibilidad dentro de una red.

En una red no dirigida, se calcula contando el número de enlaces en el camino más corto entre los nodos. En una red dirigida, se deben considerar las direcciones de las conexiones en el cálculo de la distancia geodésica. En el caso de una red pesada, se tendrá en cuenta también el peso de los enlaces para determinar la menor distancia.

Componentes conectados

Se define componente conectado como el mayor conjunto de nodos posible tal que entre cada par existe al menos un camino. Si la red es dirigida y tenemos en cuenta la dirección de los arcos, se habla de componente fuertemente conectado.

Excentricidad

La excentricidad de un nodo en una red es una medida que indica la máxima distancia geodésica entre ese nodo y cualquier otro nodo de la red.

La excentricidad es una medida útil para comprender la centralidad de un nodo en una red. Un nodo con una excentricidad baja se considera un nodo central, ya que está más cerca de todos los demás nodos y tiene una mayor influencia en la red. Por otro lado, un nodo con una excentricidad alta se considera periférico, ya que está más alejado de otros nodos y puede tener una influencia limitada en la red.

Diámetro

El diámetro de una red va a ser la distancia geodésica de los dos nodos más alejados de la red. Otra forma de definir este concepto teniendo en cuenta la excentricidad sería: “el diámetro es el valor máximo del conjunto de excentricidades de la red”.

Radio

Así como el diámetro se define como la máxima de las excentricidades, el radio se va a definir como el valor mínimo de las excentricidades.

C.5 Medidas de clustering

Coefficiente de *clustering* local

El coeficiente de *clustering* local (6) es una medida utilizada para evaluar la tendencia de los nodos de un grafo a formar grupos o comunidades. Se calcula para cada nodo y proporciona una medida de la agrupación local alrededor del nodo en cuestión.

Se define en función de los vecinos de un nodo. Si un nodo tiene k vecinos, entonces hay un máximo de $k * (k-1) / 2$ enlaces posibles entre ellos. Sin embargo, el coeficiente de agrupamiento mide la proporción de enlaces existentes entre los vecinos de un nodo en relación con el número máximo de enlaces posibles que indicábamos.

Para calcularlo en una red no dirigida se emplea la siguiente fórmula

$$C_i = \frac{2L_i}{k_i(k_i - 1)}$$

6

Donde k es el número de vecinos del nodo que estamos analizando y L el número total de enlaces de los nodos vecinos. Para calcular esta medida en una red dirigida bastaría con eliminar el 2 en el numerador.

El coeficiente de agrupamiento puede variar entre 0 y 1. Un valor de 0 indica que no hay conexiones entre los vecinos del nodo, mientras que un valor de 1 indica que todos los vecinos del nodo están conectados entre sí.[10]

Coeficiente de *clustering* local medio

Una medida global del coeficiente de *clustering* es el coeficiente de *clustering* local medio, el cual se calcula a partir de una media de todos los coeficientes de *clustering* locales (6) de cada nodo.

Transitividad

Aunque en ocasiones se emplea el mismo nombre para llamar al coeficiente de *clustering* local (6) que a la transitividad (7), se calculan de una forma distinta.

La transitividad (7) en un grafo se refiere a la propiedad de que, si hay una relación directa entre dos nodos y una relación directa entre uno de esos nodos y un tercer nodo, existe una alta probabilidad de que haya una relación directa entre los dos nodos restantes. Básicamente, es una medida de la tendencia de formar triángulos en un grafo.

La transitividad se puede calcular utilizando la siguiente fórmula para el coeficiente de transitividad, que es una medida estadística que indica la probabilidad de que se formen triángulos en un grafo, analizando las tríadas existentes y las tríadas posibles.

$$Transitividad = \frac{3 \times N^{\circ} \text{ de tríadas cerradas}}{N^{\circ} \text{ de tríadas abiertas}}$$

7

C.6 Medidas de centralidad, Centralidad y centralización

La centralidad es la medida que determina la importancia de un nodo dentro de la red en base a una serie de criterios. En función de qué criterios se escojan, podemos tener diferentes medidas de centralidad.

No siempre van a ser los mismos nodos los más importantes para todas las centralidades, ya que variará en función de los criterios, pero en la mayoría de los casos, los nodos más importantes para una medida también lo serán para el resto, posiblemente variando el orden eso sí.

Todas las imágenes que se muestran a continuación son ejemplos de redes obtenidas a través de la aplicación utilizando la librería de Python NetworkX.

Centralidad de grado

La Ilustración 2. Centralidad de Grado evalúa la importancia de un nodo en una red basándose en el número de enlaces que tiene ese nodo, es decir, en su grado. La centralidad de grado asigna mayor importancia a los nodos con un mayor número de enlaces.

En el caso de las redes dirigidas, existen diferentes variantes de la centralidad de grado, dependiendo de si se consideran únicamente los enlaces entrantes o salientes de un nodo, respectivamente.

Esta centralidad puede ser útil para identificar los nodos más conectados en una red y comprender su influencia en la difusión de información, el flujo de recursos o la propagación de enfermedades, entre otros fenómenos que ocurren en las redes. Sin embargo, es importante valorar que la centralidad de grado puede no tener en cuenta aspectos importantes de la centralidad, como la posición en los caminos más cortos o la influencia del grado de los nodos vecinos.

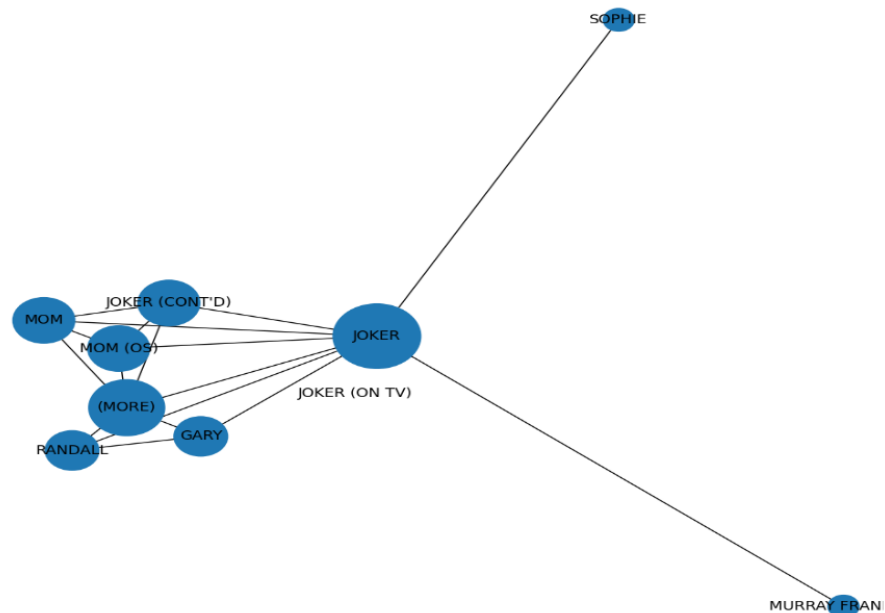


Ilustración 2. Centralidad de Grado

Centralidad de cercanía

Esta medida, evalúa la importancia de un nodo en función de su proximidad a otros nodos en la red. Se calcula determinando la longitud promedio de los caminos más cortos desde ese nodo a todos los demás nodos de la red. Cuanto más cortos sean los caminos y menor sea la distancia promedio, mayor será la centralidad de cercanía de un nodo.

La idea principal, es que los nodos con una alta centralidad de cercanía tienen una posición privilegiada en la red, ya que pueden comunicarse o transmitir información de manera más rápida y eficiente. Estos nodos actúan como "puntos de acceso" y pueden desempeñar un papel importante dentro de la red.

Sin embargo, es importante tener en cuenta que la Ilustración 3. Centralidad de Cercanía puede estar influenciada por el tamaño y la estructura de la red. En redes muy grandes o densas, es más probable que los nodos cercanos a muchos otros nodos tengan una centralidad de cercanía alta, sin llegar a cobrar la importancia que se les atribuye para toda la red.

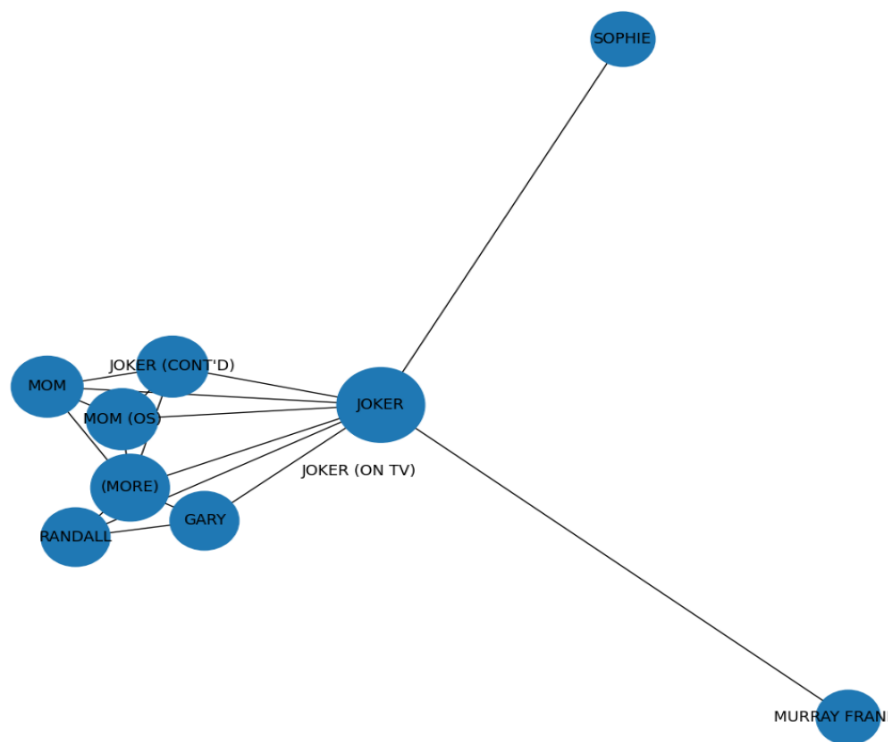


Ilustración 3. Centralidad de Cercanía

Centralidad de intermediación

La Ilustración 4. Centralidad de Intermediación valora la importancia de un nodo en base a su participación en los caminos más cortos entre otros nodos de la red, es decir, cuantifica en qué medida un nodo actúa como puente o intermediario entre otros nodos en la red.

Se calcula contando el número de caminos más cortos que pasan por un nodo en relación con el número total de caminos más cortos en la red. Un nodo con una alta centralidad de intermediación indica que es crucial para mantener el flujo de información entre diferentes partes de la red. Si se elimina un nodo con alta intermediación, la comunicación entre otros nodos podría verse obstaculizada, de ahí su importancia para mantener la conectividad en la red.

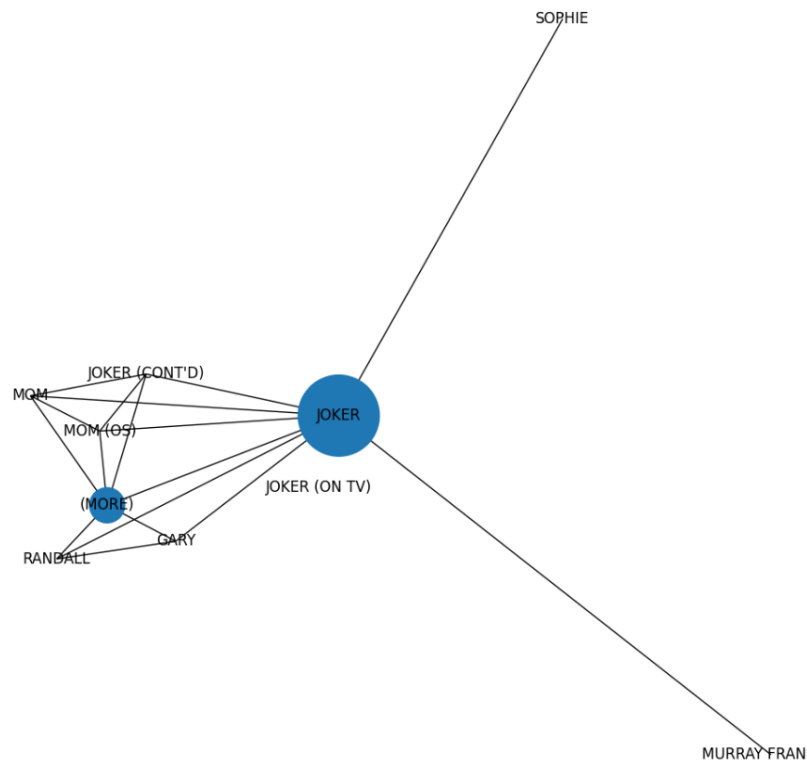


Ilustración 4. Centralidad de Intermediación

Intermediación de camino aleatorio

Esta es una variante de la medida anterior. La centralidad de intermediación se calculaba a través del número de veces que aparecía un nodo en los caminos más cortos entre dos nodos para toda la red. En este caso, no se van a seleccionar los caminos más cortos, sino que se van a seleccionar caminos aleatorios para determinar la centralidad del nodo. Aunque no se evalúen sólo los caminos más cortos, dentro de los caminos aleatorios que se establezcan para el cálculo, tendrán más peso aquellos que sean más cortos.

Centralidad de vector propio

La medida de Ilustración 5. Centralidad de vector propio, tiene en cuenta tanto la cantidad como la calidad de las conexiones que tiene cada nodo para medir su importancia dentro de la red. Según esta medida, la importancia de un nodo se deriva de la importancia de sus vecinos y de la cantidad de conexiones que tiene con esos vecinos.

Para el cálculo, se emplea un enfoque iterativo en el cual se asigna un valor de centralidad inicial a cada nodo y se actualiza ese valor en cada iteración teniendo en cuenta las conexiones con los vecinos. El proceso de iteración se repite hasta que los valores de centralidad convergen.

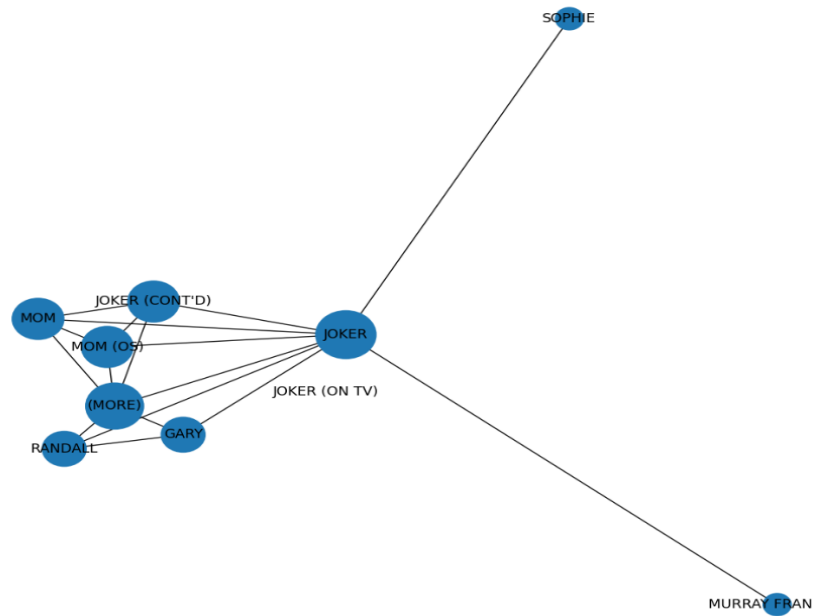


Ilustración 5. Centralidad de vector propio

PageRank

Esta medida fue desarrollada por los fundadores de Google, Larry Page y Sergey Brin, como parte del algoritmo utilizado para clasificar páginas web en los resultados de búsqueda.

La Ilustración 6. PageRank surge a partir de la centralidad de vector propio, donde la importancia de un nodo depende de los enlaces con sus vecinos y la importancia de estos, pero añadiendo alguna condición más. Principalmente, que la importancia que un nodo recibe de sus vecinos es proporcional a la centralidad de cada vecino, dividida por su grado de salida en el caso de las redes dirigidas o dividida por su número de enlaces en el caso de las redes no dirigidas.

La centralidad de PageRank ha sido muy utilizada en la clasificación de páginas web y en motores de búsqueda, ya que proporciona una medida de la relevancia de una página en función de la estructura de enlaces de la web.

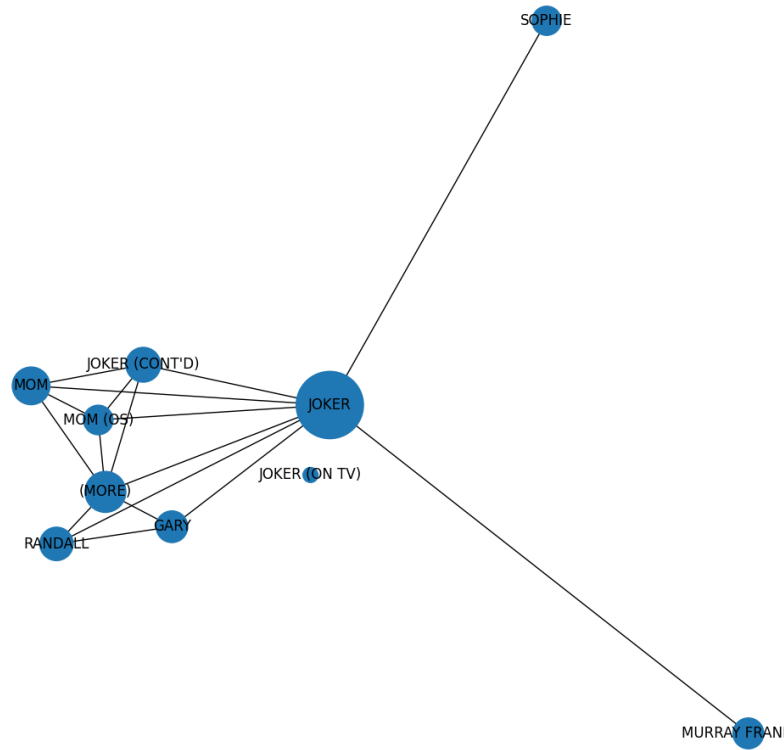


Ilustración 6. PageRank

C.7 Grupos y comunidades

La mayoría de las redes reales muestran una estructura marcadamente modular a nivel de nodos, observándose grupos de nodos densamente conectados entre sí (comunidades) o con un patrón similar de conexiones a otros nodos(clases). Cuando existe esta estructura modular, las propiedades medias de la red no se corresponden muchas veces con el promedio de cada módulo específico y se debería analizar por separado.

Un ejemplo de comunidades sencillo sería la red de conocidos de una persona, en la cual se pueden distinguir varias comunidades diferentes como podrían ser familia,

amigos, compañeros de trabajo o estudios... en función del motivo por el cual están relacionados con esa persona en concreto.

Para poder detectar los distintos grupos y comunidades que se van a formar en una determinada red, podemos emplear distintas métricas y algoritmos que vamos a ver a continuación, pero pueden surgir algunos inconvenientes:

- La definición de comunidad: “grupos de nodos densamente conectados entre sí (con escasas conexiones entre grupos)” es imprecisa, puede generar confusión, aunque también es flexible.
- Se debe buscar explícitamente estas comunidades, lo cual es un problema de cómputo complejo.

Cliques

Un clique, es un conjunto de nodos en una red que están fuertemente conectados entre sí, estableciendo un enlace entre cada par de nodos que componen esta agrupación. Por lo tanto, no hay otros nodos fuera del grupo que estén conectados al resto de miembros del grupo.

Los cliques son importantes en el estudio de las redes sociales porque al igual que las comunidades representan grupos densamente conectados de individuos que tienden a interactuar más entre sí que con otros nodos en la red, otorgando cierta información al respecto del comportamiento de los nodos. Sin embargo, la carga computacional que supone buscar cliques y decidir si ese conjunto es un clique de un tamaño concreto es muy alta y supone un problema NP-completo.[11]

K-clique

El concepto de K-Clique se puede definir de dos formas distintas en función del enfoque que se le dé. La primera definición para un K-clique, simplemente nos indicaría que es un clique conformado por K nodos.

La segunda no considera ambos conceptos como iguales y establece una diferencia entre los K-cliques y los cliques, que reside en la distancia de conexión entre los nodos del grupo. En los cliques todos los nodos que conformaban la agrupación debían estar conectados. En el caso de los k-clique, un nodo forma parte si está a una distancia (en enlaces) menor o igual que k.

En nuestro caso, para la aplicación se han empleado la segunda definición, ya que consideramos que es una métrica que otorga más información relevante.

N-clan

El enfoque del "N-clan" es una forma de abordar la detección de agrupaciones en redes donde se busca encontrar conjuntos de nodos que estén interconectados de manera más fuerte y discreta. A diferencia del enfoque "K-clique" que puede encontrar agrupaciones grandes y difusas, ya que en ocasiones puede identificar conjuntos de nodos que están conectados con nodos que no son parte del propio clique.

Para abordar esta situación, algunos analistas han sugerido imponer restricciones adicionales a los N-cliques. Por ejemplo, se puede exigir que la distancia o el camino total entre cualquier par de miembros de un N-clique cumpla con una cierta condición: asegurar que todos los enlaces entre los miembros del N-clique se realicen a través de otros miembros del mismo N-clique. [12]

K-plex

Esta condición va a permitir que los nodos sean miembros de un clique si tienen enlaces con todos excepto con otros k nodos. Este enfoque propone que un nodo forma parte de un clique de tamaño n , si tiene enlaces directos con $n-k$ nodos de ese clique. El enfoque K-plex parecer tener bastante en común con el enfoque N-clique, pero el análisis basado en K-plex a menudo arroja resultados bastante diferentes de las subestructuras del grafo. En vez de agrupaciones grandes y concatenadas, que a veces produce el análisis N-clique, el análisis K-plex tiende a encontrar números relativamente grandes de pequeñas agrupaciones. [12]

K-core

El concepto de k -core en una red se refiere al conjunto máximo de nodos en el cual, cada nodo tiene al menos grado k . Para encontrar los k -cores de una red, se realiza un proceso iterativo de eliminación de nodos que tienen un grado menor que k . Esto genera un subconjunto de nodos que forman una estructura central fuertemente conectada, puesto que cada nodo en el k -core tiene al menos k vecinos dentro del conjunto, lo que implica una mayor densidad de conexiones. [13]

K-componente

El k-componente es aquel en el que los nodos del conjunto están unidos con el resto de los nodos por un mínimo de k caminos independientes unos de otros.

Algoritmos de detección de comunidades

Para la detección de comunidades existen multitud de algoritmos, los cuales posiblemente devolverán diferentes comunidades para la misma red, en función del método de detección y criterios que apliquen.

Este proyecto emplea cuatro tipos de algoritmos de detección de comunidades que son los siguientes:

- **Girvan-Newman:**

El algoritmo de Girvan-Newman detecta comunidades mediante la detección de los enlaces más importantes para la conectividad entre comunidades. Se basa en el concepto de modularidad, que mide la calidad de la partición de un grafo en comunidades.[14]

El algoritmo está basado en el siguiente proceso iterativo:

- Inicialmente, se calcula la modularidad del grafo completo. La modularidad mide la diferencia entre el número real de enlaces que existen dentro de las comunidades y el número esperado de enlaces si se distribuyeran de forma aleatoria. Una modularidad alta indica una estructura de comunidades fuerte.
- Se identifica el enlace con el mayor valor de intermediación, que vimos anteriormente, en el grafo. Recordemos que la intermediación de un enlace, mide la cantidad de caminos más cortos entre pares de nodos que pasan a través de él.
- Se elimina el enlace seleccionado en el paso anterior y se recalcula la modularidad del grafo resultante.
- Se repiten los pasos 2 y 3 hasta que no queden aristas en el grafo o hasta que la modularidad ya no aumente significativamente.

Este proceso se repite para obtener diferentes particiones del grafo en comunidades. Una vez finalizado, se obtiene una serie de particiones y la partición con la mayor modularidad se considera la estructura de comunidades óptima para el grafo.

El algoritmo es eficaz para detectar comunidades en redes, especialmente en redes con estructuras de comunidades bien definidas. Sin embargo, puede ser computacionalmente costoso para grandes redes, ya que implica calcular la intermediación de todos los enlaces en cada iteración.

- **Louvain:**

Este algoritmo se basa en el concepto de maximizar la modularidad de una red para identificar estructuras de comunidades, al igual que el anterior.

El algoritmo de Louvain se compone de dos fases iterativas:

F.1 Fase de optimización local:

- Inicialmente, cada nodo se asigna a una comunidad diferente.
- En cada iteración, se examina cada nodo y se evalúa el cambio de modularidad resultante de mover el nodo a su comunidad vecina.
- Se selecciona el movimiento que genere el mayor incremento en la modularidad.
- Este proceso se repite hasta que no se pueda mejorar la modularidad moviendo nodos entre comunidades.

F.2 Fase de construcción de un nuevo grafo:

- En esta fase, los nodos se agrupan en "*super-nodos*", que representan las comunidades detectadas en la fase anterior.
- Los enlaces entre los super-nodos se calculan como la suma de los pesos de los enlaces entre los nodos de las comunidades correspondientes.
- Este nuevo grafo se utiliza para repetir el proceso iterativo de optimización local en la fase 1.
- Estas dos fases se repiten hasta que ya no se pueda mejorar la modularidad global.

El algoritmo de Louvain se caracteriza por ser rápido y escalable, lo que lo hace adecuado para aplicaciones en redes grandes. Además, es capaz de identificar comunidades de diferentes tamaños y detectar estructuras jerárquicas en la red.[15]

- **Clauset-Newman-Moore:**

Este algoritmo al igual que los dos anteriores, también se basa en la maximización de la modularidad de la red para identificar estructuras de comunidades.

El algoritmo CNM se compone de los siguientes pasos:

- Inicialmente, cada nodo se asigna a su propia comunidad individual.
- Se calcula la modularidad del grafo con la asignación de comunidades actual.
- Se itera sobre cada nodo y se evalúa el cambio en la modularidad si el nodo se asigna a cada una de sus comunidades vecinas.
- Se selecciona el movimiento que genere el mayor incremento en la modularidad y se actualiza la asignación de comunidades.
- Se repiten los pasos 3 y 4 hasta que no haya mejoras significativas en la modularidad o se alcance un criterio de terminación definido previamente.

Como podemos observar, este método es muy similar al método de Louvain, solo que omitiendo la segunda fase.

El algoritmo CNM es eficiente en términos computacionales y puede manejar redes grandes.[16]

- **Comunidades K-clique:**

Este algoritmo emplea el método de percolación de k-cliques, el cual emplea el concepto de k-clique que vimos anteriormente como unidades básicas de agrupación para la detección de comunidades. La percolación de k-cliques busca identificar los componentes principales y más densamente conectadas dentro de una red a través de la existencia de k-cliques.

El proceso de percolación de k-cliques se puede describir de la siguiente manera:

- Se define un valor de k , que representa la distancia de conexión máxima entre los nodos del mismo grupo.
- Se identifican todos los k-cliques en la red.
- Se construye un grafo auxiliar, llamado grafo de percolación, donde cada k-clique es representado como un nodo en el grafo.
- Se establecen enlaces entre los nodos del grafo de percolación si los k-cliques que representan tienen nodos en común.
- Se aplican técnicas de análisis de grafos para identificar las componentes principales o comunidades en el grafo de percolación. Estas comunidades corresponden a conjuntos de k-cliques que están fuertemente interconectados entre sí.

La percolación de k-cliques es especialmente útil para identificar comunidades densas y estructuras locales en redes complejas. Al centrarse en los k-cliques, se

puede obtener información sobre conexiones locales y agrupar nodos que están muy conectados en pequeñas subestructuras.[17]

C.8 Detección de roles

En este apartado se va a tratar la Ilustración 7. Gráfico de detección de roles dentro de la red. Basándonos en el algoritmo de Guimerà [18], cada nodo de la red va a tener un rol asignado, el cual es calculado mediante dos fórmulas que veremos a continuación. Para poder calcular los roles de cada nodo primero se debe haber realizado la detección de comunidades con alguno de los algoritmos que hemos visto en el punto anterior. Una vez calculadas, se identifican los roles obteniendo el grado que posee un nodo dentro de la comunidad y su coeficiente de participación. Dos nodos con características similares deberían tener el mismo rol.

Para el cálculo se van a establecer dos fórmulas:

1. El **grado del nodo dentro de su comunidad** (8), que será calculado de la siguiente forma:

$$z_i = \frac{k_i - \bar{k}_{s_i}}{\sigma_{k_{s_i}}}$$

8

- k_i : es el grado de un determinado nodo i respecto a su comunidad.
- \bar{k}_{s_i} : es la media del grado de los nodos a los que pertenece un nodo i .

2. El **coeficiente de participación del nodo** (9) de la siguiente forma:

$$P_i = 1 - \sum_{s=1}^{N_M} \left(\frac{k_{is}}{k_i} \right)^2$$

9

- k_i : es el grado del nodo i , pero esta vez respecto a la red entera.
- k_{is} : es el grado que posee un nodo i respecto a una comunidad s .

En la aplicación este apartado ha sido incluido para todos los algoritmos de detección de comunidades que hemos incorporado previamente

- Girvan-Newman.
- Louvain.
- Clauset-Newman-Moore.
- K-clique.

Cada rol aplicado viene con su propio gráfico que vamos a mostrar con la distribución de los nodos en cada uno de los roles, agradecimiento a Alicia Olivares Gil y Yi Peng Ji creadores del algoritmo que ha sido implementado para obtener este gráfico.

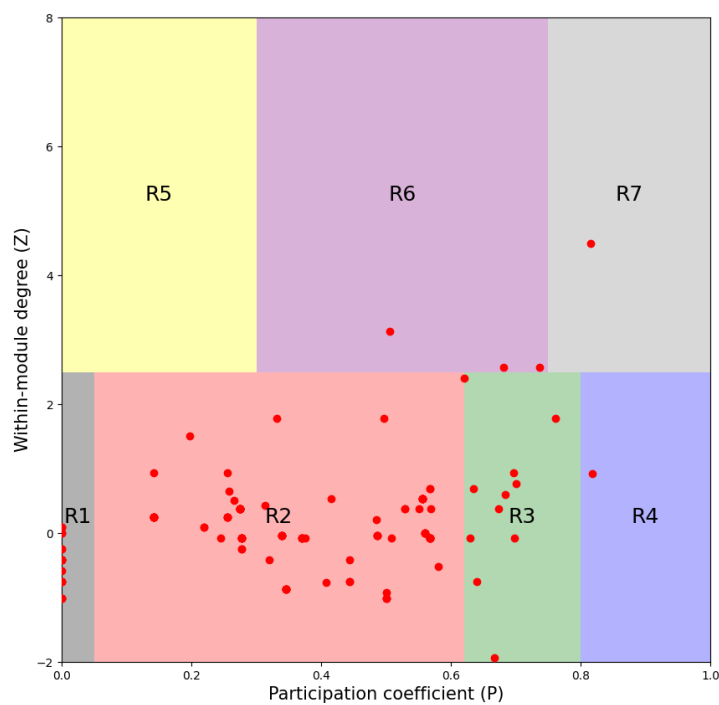


Ilustración 7. Gráfico de detección de roles

Para asignar cada nodo a su respectivo rol, una vez calculadas el grado del nodo dentro de su comunidad y el coeficiente de participación mencionadas anteriormente, se seguirá la siguiente clasificación[18] :

- Si el grado dentro de su comunidad es mayor o igual que 2.5 el nodo va a pertenecer al grupo de los **hubs** que se divide también en:
 - **Hubs provinciales (R5)**: son los nodos en los que su coeficiente de participación es menor o igual que 0.3. Poseen la característica de que 83% de sus enlaces se encuentran dentro de su comunidad.

- **Hubs conectores (R6):** son los nodos en los que su coeficiente de participación es mayor que 0.3 pero menor o igual a 0.75. Poseen la característica de que aproximadamente la mitad de sus enlaces se encuentran dentro de su comunidad.
- **Hubs kinless (R7):** son los nodos en los que su coeficiente de participación es mayor que 0.75. Son los que tienen más de la mitad de sus enlaces fuera de su comunidad y resulta muy complicado establecerles en ninguna comunidad concreta.
- Si el grado del nodo dentro de su comunidad es menor que 2.5 el nodo va a pertenecer al grupo de los **no hubs** que a su vez se va a dividir en:
 - **Nodos ultraperiféricos (R1):** son los nodos en los que su coeficiente de participación es aproximadamente 0. Poseen la característica de que tienen todos los enlaces dentro de su comunidad.
 - **Nodos periféricos (R2):** son los nodos en los que su coeficiente de participación es menor de 0.625. Poseen la característica de que tienen el 60% de los enlaces dentro de su comunidad.
 - **No hubs conectores (R3):** son los nodos en los que su coeficiente de participación es mayor de 0.625 y menor que 0.8. Poseen la característica de que tienen aproximadamente la mitad de sus enlaces dentro de la comunidad.
 - **Nodos no hubs kinless (R4):** son los nodos en los que su coeficiente de participación es mayor de 0.8. Poseen la característica de que menos del 35% de los enlaces se encuentran dentro de la comunidad. Estos nodos se encuentran mayoritariamente en modelos de crecimiento de red, pero no en redes de mundo real. Estos nodos al igual que los Hubs kinless son difícilmente asignables a una comunidad.

C.9 Scrapping Web

Hacer web scrapping, es el proceso de extraer datos de un sitio web de forma automatizada, utilizando herramientas específicas para poder acceder a la estructura HTML de una página web y extraer la información deseada.[19]

El web scraping puede ser útil en diversas situaciones, principalmente para recuperar datos para su análisis, que es el uso que le hemos dado en nuestra aplicación.

Es importante tener en cuenta que para realizar web scraping se deben respetar los términos de servicio del sitio web objetivo y asegurarse de no violar la privacidad o derechos de autor. Algunos sitios web pueden prohibir el scraping en sus términos de uso, mientras que otros pueden proporcionar una API específica para acceder a sus datos de manera legal y estructurada, como es el caso de Dracor.

C.10 Redes Dinámicas

Las redes dinámicas son un concepto que se utiliza para describir redes en las que los nodos y los enlaces entre ellos cambian y se adaptan a lo largo del tiempo, provocando variaciones en sus características en función del instante de tiempo que se analice.

Este tipo de redes se emplea en diversos campos, ya que permite observar la evolución de una red y sacar conclusiones de los motivos que han producido algún cambio en la red. Por ejemplo, en sociología, las redes de interacción dinámicas se refieren a las relaciones y conexiones entre individuos o grupos que evolucionan y se modifican en función de las interacciones sociales, permitiendo comprender cómo se propaga la información, las ideas o las influencias sociales, y cómo se forman y disuelven las relaciones entre los actores sociales.

Otro ejemplo de un campo donde cobran importancia las redes dinámicas es la biología, donde las redes de interacción dinámicas muestran las interacciones entre diferentes componentes de un sistema biológico. Estas redes pueden cambiar y reconfigurarse en respuesta a cambios ambientales o evolutivos, lo que tiene un impacto en la estabilidad y la dinámica del sistema biológico, dando información importante de en qué momento concreto se produjo la respuesta del sistema.

C.11 Obras de teatro

Una obra de teatro es la representación de un guion literario por parte de unos actores que interpretan a los personajes en un escenario, frente a un público o auditorio. La obra de teatro es el resultado de la combinación de dos formas de arte: el literario y el escénico, a través de una serie de recursos humanos y materiales.

En una obra de teatro, el texto dramático, escrito por un dramaturgo, se divide en actos y escenas que organizan la estructura de la historia. El texto contiene los diálogos y las acotaciones escénicas, que guían a los actores sobre qué decir, cómo moverse y cómo interactuar entre sí. La puesta en escena de una obra de teatro es muy importante y para desarrollarla es necesaria la selección de un elenco o conjunto de actores, la elaboración de decorados, el diseño de vestuario y algunos aspectos técnicos como la iluminación para crear la atmósfera y el ambiente adecuados para la representación.

Las obras de teatro pueden ser de diversos géneros, como comedia, tragedia, drama, farsa, musical, entre otros y abordar una amplia variedad de temas con diferentes estilos de actuación y dirección.

El teatro es una forma de arte en vivo, lo que significa que cada representación es única y puede variar en función de la interpretación de los actores y la respuesta del público. Las obras de teatro pueden ser presentadas en multitud de lugares distintos desde teatros profesionales o comunitarios hasta escuelas o en la propia calle, siempre que sean adecuados para la representación teatral.[20]

C.12 Corpus Lingüístico

Un corpus lingüístico es una colección organizada y amplia de ejemplos reales de uso del lenguaje, ya sea en forma de textos escritos o muestras orales transcritas. Los corpus pueden ser textuales o de referencia, y su objetivo es reflejar de manera precisa y realista el idioma o su variante.

Los corpus son herramientas fundamentales en el estudio del lenguaje, ya que ofrecen acceso a una gran cantidad de información lingüística y enciclopédica. No tienen una estructura cohesiva y no se analizan en su totalidad, sino que se utilizan como base para investigaciones lingüísticas. Su recopilación y análisis eficiente ha impulsado el desarrollo de la lingüística computacional, provocando que hoy en día, los corpus se recopilen y almacenen electrónicamente para su posible investigación.

En nuestro caso tendremos a nuestra disposición corpus de obras de teatro, que realmente serán agrupaciones de obras en base a un criterio común, como puede ser su autor o su lugar de procedencia.

D.TÉCNICAS Y HERRAMIENTAS

En este apartado vamos a explicar las técnicas y herramientas que han sido empleadas durante el desarrollo del proyecto

D.1 Metodología ágil – Scrum

La metodología SCRUM es un enfoque ágil de gestión de proyectos que se utiliza comúnmente en el desarrollo de software, aunque también se puede aplicar a otros tipos de proyectos. Scrum se centra en el desarrollo incremental y la adaptación continua.

En Scrum, un proyecto se divide en ciclos de trabajo llamados "sprints". Cada sprint tiene una duración fija, durante la cual se desarrolla un conjunto de funcionalidades del proyecto. Al final de cada sprint, se realiza una revisión de todo el trabajo realizado, para valorar los objetivos cumplidos y establecer los siguientes de cara al siguiente sprint.

Para un control del trabajo y valoraciones de objetivos razonables, se realizan reuniones diarias de trabajo y estimaciones de tiempo para cada tarea.[21]

D.2 Herramienta de control de versiones

- **Herramienta:** GitHub

GitHub es un portal que nos permite albergar nuestros proyectos empleando el control de versiones que ofrece Git.[22]

Ha sido elegida debido a su sencillez de uso comprobada en la utilización previa en alguna asignatura de la carrera. También por la integración de ZenHub[23], una herramienta que será explicada posteriormente. Otro motivo ha sido que las versiones anteriores ya estaban alojadas en esta plataforma y nos servían como guía y ejemplo.

- **Alternativa:** Gitlab

Se evaluó la posibilidad de emplear Gitlab al estar más familiarizado con esta plataforma, pero finalmente se descartó la idea al tener menos motivos de uso que Github.

D.3 Herramienta de gestión del proyecto

- **Herramienta:** ZenHub

ZenHub es una herramienta de gestión de proyectos y seguimiento de tareas diseñada idealmente para equipos de desarrollo de software. Una de las características más destacadas de ZenHub es su capacidad para mostrar información adicional directamente dentro de GitHub. Los usuarios pueden ver los datos de ZenHub en el tablero Kanban y en las páginas de los problemas y las solicitudes de extracción de GitHub, lo que facilita la planificación y el seguimiento de los proyectos sin tener que cambiar constantemente de herramienta.[23]

D.4 Herramienta para realización de la documentación

- **Herramienta:** Microsoft Word

Se ha decidido emplear la herramienta de tratamiento de texto Word debido al conocimiento que ya se tenía previo al proyecto, aunque no fue la primera decisión como explicaremos en las alternativas.

Alternativas: En un principio se iba a desarrollar la memoria con el sistema de composición de textos LaTeX y así está indicado en la toma de decisiones al comienzo del proyecto. Sin embargo, la curva de aprendizaje requerida para el desarrollo de la documentación suponía un problema con los tiempos de entrega restantes. Había interés en aprender a manejar esta herramienta, pero no suponía ninguna ventaja respecto a Word al no haber gran cantidad de fórmulas en la documentación y podía complicar la entrega del proyecto.

D.5 Herramienta para gestión y edición del repositorio local-remoto

- **Herramienta:** Visual Studio Code

Visual Studio Code es un editor de código fuente ampliamente utilizado por los desarrolladores debido a su interfaz intuitiva, soporte para múltiples lenguajes y tecnologías, capacidades de edición y depuración avanzadas, extensibilidad y personalización, así como su integración con herramientas externas como el control de versiones de git. Debido a que reunía todas estas características, ideales para el desarrollo del proyecto fue la elección final. El ser gratuito también tuvo cierta importancia. [24]

D.6 Herramienta para la gestión de referencias bibliográficas

- **Herramienta:** Mendeley

Como la intención inicial era haber desarrollado la memoria con LaTeX, el gestor de referencias elegido fue Mendeley ya que ofrecía mejores características para este desarrollo. Es una plataforma que permite organizar, almacenar y compartir referencias bibliográficas y artículos científicos. Se pueden importar referencias, leer y anotar documentos, generar citas y bibliografías automáticamente.[25]

- **Alternativa:** Zotero

Zotero es otro gestor de referencias muy similar a Mendeley. Tal vez para el uso de Word tiene una mejor implementación y también una extensión para navegadores que hubiera facilitado la adición de referencias, pero como ya hemos mencionado la elección se realizó al comienzo y al tener almacenadas ya referencias no hubo cambio en la decisión.

D.7 Lenguaje de programación

- **Herramienta:** Python

Los conocimientos previos tras su manejo en asignaturas de la carrera, que las versiones previas del proyecto estuvieran ya desarrolladas así y ser uno de los lenguajes de programación con mejores herramientas para el tratamiento de redes y grafos hacen que Python sea el lenguaje de programación elegido sin alternativas.

D.8 Interfaz gráfica

- **Herramienta:** Flask

Flask es un marco de desarrollo web de código abierto en Python, utilizado para crear aplicaciones web y APIs de forma rápida y simple. Flask fue desarrollado para ser fácil de usar sin imponer una estructura rígida a las aplicaciones, lo que brinda mucha flexibilidad al desarrollo.

Una de las características clave de Flask es su capacidad para integrarse con otras bibliotecas y marcos de Python. Por ejemplo, está basado en la especificación WSGI (Web Service Gateway Interface) de Werkzeug y el motor de plantillas Jinja2 [26].

D.9 Herramienta para la interfaz

- **Herramienta:** Bootstrap4

Bootstrap es un marco de trabajo para desarrollo web que proporciona un conjunto de herramientas y estilos predefinidos para la creación rápida y sencilla de sitios web.

Una de las principales ventajas de Bootstrap es su enfoque en el diseño responsivo, lo que implica que los sitios web creados con Bootstrap se adaptan automáticamente a al tamaño de pantalla del equipo donde se está visualizando la página, permitiendo al usuario tener una experiencia óptima sin la necesidad de realizar muchos cambios adicionales.

Bootstrap incluye una amplia variedad de componentes y estilos CSS predefinidos, así como un conjunto de plantillas que facilita la creación de diseños de página flexibles y organizados, siendo este el motivo de su elección en versiones previas.[27], [28]

En concreto para esta versión, se ha decidido mantener puesto que es la elección ideal y por su conocimiento y manejo previo de esta herramienta para diseños html.

D.10 Programación en el lado del cliente

- **Herramienta:** JavaScript + jQuery

JavaScript es un lenguaje de programación interpretado y orientado a objetos utilizado principalmente en el desarrollo web. Permite agregar interactividad y dinamismo a las páginas web al manipular el contenido y estilo de elementos HTML a través del DOM, controlar eventos y realizar operaciones matemáticas. Es una parte esencial del desarrollo web moderno y cuenta con una amplia gama de bibliotecas y frameworks que facilitan el desarrollo de aplicaciones más complejas.[29]

jQuery es una de esas bibliotecas que mencionamos de JavaScript, que simplifica la manipulación de elementos HTML, el manejo de eventos, la animación y el uso de AJAX en el desarrollo web. Proporciona una sintaxis muy fácil de usar y concisa, permitiendo seleccionar y modificar elementos del DOM rápidamente. [30]

Ambas elecciones, al igual que en otras herramientas, se mantienen las mismas que en las versiones anteriores.

D.11 Tablas con filtro

- **Herramienta:** DataTables

DataTables es un complemento para la biblioteca jQuery. Es una herramienta altamente flexible, construida sobre las bases de la mejora progresiva, que agrega todas estas funciones avanzadas a cualquier tabla HTML. Facilita un gran número de herramientas para el manejo de desarrollo de tablas, permitiendo añadir opciones como filtrados, traducciones o diseños responsive.[31]

D.12 *Scraping* Web

- **Herramienta:** BeautifulSoup4 + urllib

BeautifulSoup4 es una librería de python que permite la obtención de forma sencilla de información de una página web, haciendo una búsqueda por un tipo de etiqueta html concreto.

Urllib es una librería de Python para la gestión y trabajo con direcciones web. En el proyecto se ha empleado el paquete “urllib.request” para establecer conexión con la dirección web requerida y la librería BeautifulSoup4 para extraer la información de dicha dirección web.

D.13 Herramienta para la obtención de datos de las obras de teatro

- **Herramienta:** Dracor

DraCor es la abreviatura de "drama corpora", un proyecto que contiene una creciente colección de obras en (en su mayoría) idiomas europeos, todas codificadas en TEI. Dispone de una API la cual permite obtener la información de todas las obras y corpus de la colección para su manipulación y ha sido el eje de trabajo de esta versión 3.0. [32]

D.14 Funcionalidad etnia y sexo

- **Herramienta:** Ethena + Genni

Ethnea y Genni es una herramienta empleada para obtener una predicción del sexo y la etnia de una persona a través del nombre y apellido de una manera sencilla y rápida. Se realiza scraping, dado que no está disponible como una herramienta para Python, permitiendo obtener así el sexo y la etnia del personaje, debido a que “Ethnea + Genni”

permite obtener la información en formato json simplemente añadiendo “&format=json” a la url.[33]

D.15 Generación de la red estática

- **Herramienta:** NetworkX

NetworkX es una biblioteca de Python utilizada para la creación y análisis de redes complejas. Permite construir y manipular grafos dirigidos o no dirigidos, visualizar las redes y realizar análisis mediante una variedad de algoritmos. Es una herramienta versátil que facilita el estudio de la estructura y las relaciones. NetworkX se integra con otras bibliotecas populares de Python, aunque para algunas características en concreto ha sido necesaria la instalación de una librería adicional complementaria: Python-louvain.[34], [35]

También hay que mencionar que ha sido empleada para la exportación de la red estática a diferentes formatos como pueden ser GEXF o GML, gracias a las funciones que ofrece.

Igual que mencionábamos para Python, su uso en otras asignaturas ha ayudado en su elección y manejo.

D.16 Generación de la red dinámica.

- **Herramienta:** DyNetX

DyNetx [36] es una biblioteca de Python dedicada al análisis de redes dinámicas y construida sobre NetworkX. Permite crear, manipular y analizar redes que cambian en el tiempo. Sus características incluyen gráficos dinámicos, atributos temporales en las aristas, métricas dependientes del tiempo, caminos temporales y visualización de redes dinámicas.

D.17 Visualización de la red

- **Herramienta:** network styling with d3

Para la visualización de la red se ha empleado la biblioteca de javascript D3, la cual permite a partir de datos, generar infogramas dinámicos e interactivos en navegadores web, con distintas opciones para modificar la red a gusto del usuario. Teóricamente, la aplicación web te pide que introduzcas una dirección, pero para ahorrar

trabajo repetitivo al usuario, la aplicación introducirá directamente la red de forma automática.[37]

- **Alternativas:** Visualizar la red con networkX, tal y como se hace en los informes. Esto realmente muestra una imagen fija, que no permite al usuario modificar los parámetros ni visualizar de forma dinámica, limitando considerablemente esta funcionalidad. Otra opción contemplada fue emplear netwulf, que internamente emplea también d3, debido a que dracor ofrece tutoriales para la integración de su API directamente en esta librería. Fue descartado porque no ofrecía ninguna ventaja una vez teníamos construida la red.

D.18 Herramienta para la exportación a Gexf Dinámica

- **Herramienta:** Gexfpy

Así como la exportación estática mencionábamos que la realizábamos con NetworkX [35], esta librería no nos permite realizar exportaciones dinámicas.

Para ello hemos empleado GexfPy [38], que es una librería de Python que permite la creación de una forma sencilla de un fichero GEXF. Solo debemos ir pasándole las etiquetas del fichero gexf que queremos que aparezca y nos devolverá la construcción completa del árbol de propiedades para la red dinámica.

He de mencionar también, que para comprobar la correcta exportación de todos los ficheros GEXF se ha empleado la aplicación Gephi [39].

D.19 Herramientas para la exportación a vídeo

- **Herramienta:** matplotlib Animation

La librería Matplotlib[40] es una librería de Python que permite al usuario visualizar datos de diferentes formas a través de gráficos y representaciones. En concreto, su módulo animation[41], nos permite crear animaciones a partir de representaciones de figuras que se unirán para formar una animación completa, que en nuestro caso, serán los instantes de la evolución de la red. El módulo permite establecer la calidad y velocidad de reproducción, así como el intérprete para realizar una exportación a vídeo.

- **Herramienta:** FFmpeg

FFmpeg es una utilizada para grabar, convertir y transmitir audio y video. Ofrece una amplia gama de funcionalidades relacionadas con el procesamiento multimedia, como la captura de video y audio desde cámaras y micrófonos, la conversión de formatos de archivos multimedia, la edición y recorte de videos, la extracción y adición de pistas de audio, y muchas otras tareas relacionadas con el manejo de medios [42]. Será la empleada para convertir la animación generada por la librería Matplotlib.Animation [41] en formato MP4 y tener así la animación en formato vídeo para una cómoda visualización.

D.20 Realización de la wiki

- **Herramienta:** Wikidot

Wikidot es una plataforma en línea que permite crear sitios web colaborativos, especialmente wikis. Proporciona herramientas de edición y colaboración sencillas, así como funcionalidades adicionales como foros y permisos de usuario. Permite la creación de hasta cinco wikis de manera gratuita con la limitación de ser de un tamaño limitado, pero para el desarrollo que queremos realizar nos sobra. [43]

D.21 Herramienta para gestionar las traducciones

- **Herramienta:** Poedit

Para la realización de la internacionalización, se ha empleado el editor de traducciones Poedit[44], el cual permite una fácil internacionalización, gracias a la importación de un fichero. Pot, generado con la extensión Flask-Babel que mencionábamos anteriormente[45]. Con este fichero cargamos todos los textos detectados según nuestra configuración que queremos traducir y bien manualmente o mediante la opción automática que nos ofrece Poedit realizamos la traducción al idioma deseado.

D.22 Herramienta para generación de diagramas

- **Herramienta:** Draw.io

Draw.io es una aplicación que te permite crear fácilmente una amplia variedad de diagramas UML de manera cómoda. Además, cuenta con una colección de plantillas que pueden ser útiles para crear estos diagramas. Es gratuita y no requiere descargar ningún

software adicional. Simplemente al acceder al enlace proporcionado, puedes comenzar de inmediato a crear tus diagramas UML.[46]

D.23 Herramienta para la obtención de gráficos de sprint

- **Herramienta:** EasyRetro.io

Debido a problemas con la licencia de ZenHub no se ha podido realizar el seguimiento correctamente de las diferentes tareas y sprint. Por ello se ha intentado buscar herramientas alternativas para al menos poder ver la evolución de los sprints.

Es el caso de EasyRetro.io [47], que permite realizar funcionalidades parecidas a ZenHub [23]. Ha sido empleada para generar los grafos BurnDown que muestran la evolución de trabajo para los diferentes sprints y poder visualizar mejor la planificación del proyecto.

D.24 Herramienta para albergar la aplicación

- **Herramienta:** Railway

Railway es una plataforma para poder desplegar aplicaciones de diferentes tipos en la nube, desde una base de datos hasta una aplicación web. En principio es de pago, pero ofrece un plan mensual gratuito con valor de cinco dólares al registrarnos con github, lo cual permite tener una aplicación desplegada con hasta quinientas horas de uso al mes.[48]

Es muy similar a Heroku, incluso en su documentación lo mencionan, con un despliegue sencillo para no tener que preocuparse de ello.

Como el resto de las aplicaciones de este tipo, se pueden aumentar las prestaciones pagando por ello.

E. ASPECTOS RELEVANTES DE DESARROLLO DEL PROYECTO

En este apartado se van a tratar todos aquellos aspectos de interés relativos al desarrollo desde el punto de partida. Se explicarán las nuevas implementaciones que se han añadido, los errores corregidos y retoques aportados hasta el despliegue final junto con todas las dificultades que han ido apareciendo.

E.1 Inicio del proyecto

A la hora de elegir temática para el trabajo de fin de grado, la idea era realizar un trabajo relacionado con alguno de los temas que más me habían llamado la atención durante la carrera como podían ser los sistemas inteligentes, minería de datos o redes de interacción, buscando profundizar más en mi trabajo y conocimientos en alguno de estos campos de cara a mi formación futura.

Tras valorar las opciones disponibles y hablar con los tutores, me decanté por este proyecto ya que me llamaba la atención el desarrollo de redes de interacción con ejemplos reales de novelas, películas y obras de teatro y el ser una tercera versión de un proyecto ya existente me quitaba de muchas tomas de decisiones que no me interesaban tanto.

Una vez formalizada y aprobada la petición, comenzó el desarrollo del trabajo.

E.2 Metodologías

Se ha tratado de aplicar una metodología SCRUM en la medida de lo posible, pero debido a las condiciones del proyecto y a una serie de problemas durante el desarrollo, no se ha podido aplicar al cien por cien, existiendo algunas diferencias respecto a la idea inicial:

- Se ha diseñado la aplicación de una forma incremental
- Se ha intentado mantener un tiempo para cada sprint de dos semanas, sin embargo, hubo un par de meses en los que por motivos personales no se pudo tener reuniones ni avanzar lo esperado en el sprint y en las últimas semanas debido a la proximidad de la fecha de entrega, se ha realizado algún sprint en el plazo de una semana.
- Se han mantenido reuniones al principio de cada sprint con el fin de establecer las funcionalidades que se debían añadir o corregir.

- Se han realizado revisiones al final de cada sprint con el objetivo de ver los avances y corregir posibles errores de cara al siguiente sprint.
- La idea era emplear la herramienta de ZenHub, para realizar el control y gestión de sprints, pero debido a un error con la creación del repositorio inicial y un problema con la duración de la licencia del alumno no se ha podido utilizar ni buscar una alternativa decente puesto que ya se habían realizado bastantes avances y quedaba poco trabajo de desarrollo.

E.3 Formación

Para algunas herramientas no ha sido necesaria la formación, como puede ser el ejemplo de lenguaje de desarrollo *Python*, empleado en varias asignaturas de la carrera o el marco de trabajo *Bootstrap*, empleado por el alumno en su trabajo habitualmente. Tampoco ha sido necesaria la formación en alguna de las herramientas del proyecto que cuyas funcionalidades se empleaban principalmente en las versiones anteriores y al realizar solamente pequeñas modificaciones no ha sido necesario un manejo avanzado. Sin embargo, si que ha habido algunas herramientas que han requerido de una formación previa, para su correcto manejo e implementación.

Dracor API, NetworkX y Netwulf

Para la recuperación de los datos necesarios para el módulo de obras de teatro, primero hubo que realizar una serie de pruebas y revisar algunos ficheros y ejemplos que ofrece la plataforma a modo de tutorial en los siguientes enlaces:

- <https://github.com/dracor-org/dracor-notebooks/tree/main/api-tutorial>[32]
- <https://dracor.org/doc/api> [32]

Una vez realizadas varias pruebas, primero se implementó la obtención de los datos básicos para la selección de los corpus y de las obras y más adelante se implementó la recuperación de los datos para los personajes de las obras seleccionadas.

En el caso de la obtención de la red estática hubo más complicaciones. Al ofrecernos *Dracor* la posibilidad de obtener directamente la red estática ya formada, era un coste innecesario construir la red a partir de los datos obtenidos, pero esto supuso un problema por el formato en el cual se devolvía la red y su conversión al formato empleado, que derivó en tener que investigar posibles soluciones.

Una de las soluciones implicaba a las librerías de *NetworkX*, las cuales hubo que analizar el código y entender como funcionaban, para valorar añadir una posible modificación al código que nos permitiera realizar la exportación de los datos que nos devolvía *Dracor* al formato de red que empleaba *NetworkX*.

La otra solución pasaba por no realizar la conversión a *NetworkX* y convertirlo a la librería que emplea la visualización de la red. Esta solución surge porque uno de los tutoriales de *Dracor* aborda el tratamiento de las redes con esta librería [49].

Hubo que valorar ambas opciones y para ello formarse previamente sobre cual era la más conveniente y cual nos iba a suponer una mayor carga de trabajo. Finalmente se realizó la implementación que veremos a continuación, que ofrecía buenos resultados y no supuso mucha carga, descartando la opción de *Netwulf*.

Datatables

Para la selección de los corpus y las obras se buscó realizar una implementación similar a la que presenta *Dracor* y para ello se seleccionó el complemento *DataTables* [31] para *jQuery* que nos permitía exactamente mostrar los datos de obras y corpus en una tabla filtrable.

Para el manejo de *DataTables* hubo que consultar en varias ocasiones su documentación, ya que hubo que realizar varias adaptaciones.

Primero añadimos una tabla sencilla sin mucho desarrollo, con *scroll* lateral, eliminando el paginado y mostrando el número de registros, basándonos en la documentación básica: https://datatables.net/examples/basic_init/

El siguiente paso fue añadir un filtrado y una ordenación por columnas, para permitir al usuario poder buscar fácilmente algún registro concreto, de manera similar a como se presenta la información en *Dracor*. Para ello revisamos la documentación del siguiente enlace: https://datatables.net/examples/data_sources/dom.html

Al realizar la internacionalización nos encontrábamos con que las opciones de paginado y búsqueda para los filtros no seguían el mismo criterio que habíamos aplicado con *Flas-Babel* [45], así que tuvimos que revisar otra vez la documentación para buscar que posibilidades había. <https://datatables.net/plug-ins/i18n/Spanish>

Por último, para tratar de mantener la aplicación con el diseño responsive otorgado gracias al marco de trabajo de Bootstrap [28], había que tratar de adaptar las tablas y para

ello otra vez tocó recurrir a la documentación de *DataTables*: <https://datatables.net/extensions/responsive/>

E.4 Desarrollo de nuevas implementaciones

Módulo Obras de teatro

Como ya se ha mencionado en el apartado de formación, la primera implementación fue permitir la selección y visualización de obras de teatro y adaptar todas las funcionalidades ya existentes a este nuevo módulo.

Lo primero fue permitir la selección del corpus de obras para posteriormente seleccionar la obra deseada. Para ello a través de las tablas de *DataTables* se muestra todos los corpus disponibles en *Dracor* y se permite la búsqueda y filtrado para otorgar una selección más sencilla. De manera similar se permite la selección de obras. Para ambas tablas se muestran aquellos campos que pueden mostrar información más interesante para el usuario como puede ser el título o el número de obras disponibles para los corpus o el autor y el año de publicación para las obras.

Una vez seleccionada la obra se obtendrán a través de la API todos los personajes correspondientes para dar paso al menú de edición. En este menú se han ocultado varias de las funcionalidades que se implementaban anteriormente para las películas o las novelas, como pueden ser la adición de personajes o la modificación del Id, puesto que al obtener todos los datos de *Dracor* no tiene sentido permitir al usuario que pueda editar o añadir este tipo de datos, los cuales van a ser fijo e inmutables.

Para la obtención de las apariciones se vuelve a emplear la API de *Dracor* que para cada personaje nos devolverá sus apariciones, junto con alguna métrica extra como puede ser el género del personaje. Es por eso que para este menú de selección se ha separado la opción de calcular etnia y sexo, ya que para el segundo no hará falta emplear la herramienta de *Ethnea* y *Genni*.

Los siguientes cambios serán para la obtención de las redes que se explicará en el punto siguiente

Redes para obras de teatro, función `Read_Gexf`

Como la propia API nos devuelve la red completa era un poco absurdo calcularla enlace a enlace como se hacía para películas y novelas. El problema que surgió fue que la API nos devolvía la red en formato GEXF.

La librería *NetworkX* tiene una función que permite la lectura y conversión en este tipo de formato, pero solo a través de un fichero y no de una cadena de caracteres como era el caso.

Es por eso por lo que se plantearon diversas posibilidades, desde guardar un fichero temporal en la sesión del usuario, hasta modificar la función para permitir la entrada por cadena o incluso cambiar la visualización para emplear *Netwulf* en lugar de *NetworkX* a la hora de construir la red.

Finalmente se optó por modificar la función de *Read_Gexf* y permitir que la entrada fuera por cadena en lugar de por fichero, lo cual solo supuso cambiar un par de métodos de lectura dentro de la propia función.

Para la visualización dinámica de las obras de teatro no se pudo obtener directamente a través de la API ya que *Dracor* no contempla las redes dinámicas, sin embargo si que nos devuelve los personajes que intervienen en cada escena de la obra seleccionada, así que en un nuevo método, de manera similar a como se realizaba para novelas y películas, recorreremos los personajes de cada escena, estableciendo enlaces entre todos aquellos personajes coincidentes o actualizando pesos de los enlaces ya existentes.

Mejoras gráficas y visuales

Se han implementado diferentes mejoras visuales respecto a las versiones anteriores, para tratar de ofrecer una interfaz más sencilla y fácil de usar para el usuario.

- Se ha eliminado la opción “*Obtener posiciones*” que aparecía en el menú de personajes y se ha dejado solo la opción de “*obtener etnia y sexo*” / “*obtener sexo*”. Esta opción al estar marcada obtenía las apariciones de los personajes. Después de revisar su funcionamiento y ver que no aportaba nada a mayores y por tanto era redundante, se ha eliminado y ahora las apariciones se calculan automáticamente siempre que algún personaje no tenga un número de apariciones asignado.
- En las pantallas donde se visualizaban las redes, se dejaba mucho espacio en los márgenes, lo cual en un monitor pequeño dificultaba el manejo y visualización de la red. Se ha disminuido el tamaño de los márgenes y centrado la red para una mejor visualización.
- Se ha modificado los textos para la red dinámica, ya que antes la medida de tiempo que se mostraba para la red y el informe dinámicos era intervalo,

lo cual no es correcto, ya que técnicamente visualizamos y analizamos instantes de tiempo.

- Se ha sustituido la entrada de instantes de tiempo concretos para su visualización. Antes se introducía un número, sin realizar ninguna validación y se hacía clic en el botón de buscar. Ahora la entrada está limitada con un deslizable, que va desde el instante inicial hasta el final. Se sigue teniendo que hacer clic en botón.
- Se han limitado los botones anterior y siguiente cuando se alcanzan los límites de tiempo inicial y final respectivamente, para no intentar visualizar instantes de tiempo inexistentes.
- Se han modificado los parámetros de la exportación a vídeo para una mejor visualización, limitando los frames por segundo, el tamaño de los nodos y las etiquetas de los personajes.
- Se ha retocado las cabeceras de las métricas relativas a la detección de comunidades, ya que solo se mostraba en la cabecera el número de comunidades del instante final.

Nuevas implementaciones dinámicas

- Exportación GEXF:

La exportación a GEXF que se realizaba anteriormente para las gráficas dinámicas no era realmente dinámica, pues se hacía el mismo tratamiento que para la red estática.

Para poder realizar la exportación dinámica primero se ha consultado el siguiente manual [50], para ver como es el formato de dinámico de un fichero GEXF. Este manual explica que para que un fichero sea considerado dinámico se debe establecer el modo del grafo a dinámico y se deben dar una serie de etiquetas de inicio y final a todos elementos que se quieran modificar dinámicamente, nodos, enlaces y pesos en este caso.

Para construir el fichero GEXF se ha empleado la librería GexfPy [38] la cual nos permite mediante una serie de modelos, ir construyendo la estructura del fichero GEXF paso a paso, e ir añadiendo todos los elementos que requiere el fichero dinámico.

Ya de manera similar a como se ha realizado la red dinámica, teniendo todos los enlaces de la red con los instantes de tiempo en los que se forman, se crea el fichero GEXF dinámico.

- Ejecución automática:

Como la ejecución instante a instante puede ser tediosa para el usuario y las exportaciones limitan uno de nuestros objetivos que es el crear una herramienta que ofrezca un fácil manejo de redes de interacción, se ha implementado una opción de ejecución automática, para que el usuario pueda ver la evolución de la red sin tener que estar metiendo valores a mano o exportando ficheros.

Para ello se ha añadido un botón, en el menú de visualización dinámica, que en el momento en que se haga clic, comienza la ejecución automática, avanzando directamente al siguiente instante de tiempo respecto al actual y avanzando instante a instante automáticamente cada 2 segundos hasta al alcanzar el último, momento en el cual la ejecución se detendrá.

El usuario puede detener la ejecución automática en cualquier momento haciendo clic en el mismo botón, el cual en función del estado de la ejecución cambiará su texto, en este caso mostrando el mensaje “DETENER”, o bien haciendo clic en cualquier otra funcionalidad.

Hay que recalcar que la ejecución comienza a partir del instante actual, avanzando directamente hasta el siguiente instante, sin tiempo de espera. Esto se ha realizado así porque una ejecución automática entera de principio a fin sin posibilidad de parar limitaba bastante y añadir un tiempo de espera desde que se hacía clic hasta que avanzaba podía provocar errores.

E.5 Corrección de errores

En este apartado se va a hacer mención a la corrección de errores, tanto propios que han ido surgiendo durante el desarrollo, como de versiones anteriores, como errores surgidos por versiones de librerías desactualizadas.

Errores por librerías desactualizadas

Debido a la actualización de algunas librerías para la incorporación de nuevas funcionalidades, algunas de las funcionalidades que ya estaban implementadas han dejado de funcionar o han alterado su funcionamiento, así que ha habido que realizar ajustes.

En la mayoría de los casos, con un par de búsquedas se ha encontrado el error, que suele ser la modificación de algún parámetro en la llamada a funciones de la librería

actualizada. Esto se ha solucionado de manera sencilla, corrigiendo los parámetros necesarios, como ha sido en el caso de las funciones de edición de personajes o las exportaciones, las cuales devolvían errores de este estilo.

El otro error importante que ha aparecido ha sido en la exportación de las animaciones dinámicas. Aparecía un mensaje de error y buscando información, se llegó a la conclusión de que el problema estaba en el PATH de la librería, que no referenciaba correctamente al intérprete FFmpeg[42] para la exportación de vídeo. En versiones anteriores se empleaba la librería para Python de FFmpeg_Python, que era una adaptación de este intérprete, pero no se sabe porque ya no funciona. La solución ha sido incluir en el proyecto el ejecutable correspondiente y referenciarlo correctamente para evitar que ocurra este error. Como el fichero era muy grande, se ha subido comprimido y habrá que indicar en los ficheros de instalación que para una correcta ejecución se debe descomprimir.

Errores propios y refactorización

A lo largo del desarrollo fueron apareciendo diversos errores, sobre todo al no considerar todos los casos posibles en la incorporación de las obras de teatro. Es por eso que aparecieron errores al obtener las redes cuando se borraba algún personaje o no se consideraron correctamente los instantes iniciales y por ello aquellas obras de teatro que solo disponían de una escena aparecían vacías.

También se tuvo que añadir un tratamiento especial para aquellos nodos que aparecen en alguna escena, pero que no tienen enlaces. Esto es debido a que en la obra un personaje realiza un monólogo, sin intervenir con otros personajes, afectando sobre todo a nuestra forma de construir la red, basada en los enlaces existentes y por ello hubo que realizar modificaciones, tanto para la generación de la red como para las diversas exportaciones que realizamos.

Una vez desarrollada la obtención de las redes para las obras de teatro, se realizó una refactorización del código, puesto que la obtención de los datos se realizaba en dos pasos diferentes, de distinta manera a como se realizaba para novelas y películas, donde en el mismo paso se obtenía los datos para todos los personajes para ya después formar la red estática o la red dinámica según corresponda. Se adaptó para poder realizarlo de la misma manera y así emplear los mismos métodos que se utilizaban anteriormente y unificar lo máximo posible el código.

Errores de versiones previas

Se han corregido diversos errores y realizadas ciertas mejoras en aspectos del código que tal vez no eran los óptimos en versiones anteriores.

Al realizar la refactorización y unificar todo en el mismo método, surgieron varios errores respecto al tratamiento de enlaces, por estimaciones que no se estaban realizando o que no eran las correctas. Por ejemplo, a la hora de evaluar los enlaces existentes entre personajes, todos los enlaces se generaban duplicados y posteriormente se tenían que estar descartando. Descarte el cual no se realizaba del todo correcto y eliminaba algunos enlaces que no debía. Se ajustaron ambos errores para mejorar el tratamiento y la eficiencia.

Tampoco se realizaba el tratamiento correcto para los nodos aislados sin enlaces que se mencionaba en el punto anterior. Se incorporó de igual manera que para las obras de teatro.

Por último, la obtención de las métricas del informe dinámico no era del todo óptima, puesto que primero se obtenían todas, independientemente de cuales hubiera seleccionado el usuario, y posteriormente solo se mostraban aquellas seleccionadas. Se corrigió para solo calcular las métricas seleccionadas y ganar en eficiencia.

E.6 Internacionalización

El proyecto no se encontraba del todo internacionalizado respecto a la última versión y algunos textos aparecían sin traducir. Esto sumado a las nuevas adiciones que se habían realizado.

Para ello, de la misma manera que se realizó en la versión 1.0 [51] se ha realizado la internacionalización completa del proyecto, ofreciendo la posibilidad de visualizar la aplicación tanto en inglés como en español.

E.7 Ayuda al usuario

Se ha intentado facilitar el manejo y tratar de realizar la aplicación lo más sencilla e intuitiva para el usuario. Para ello se han modificado varios textos que aparecen en la aplicación, intentando dejar claras las funcionalidades, tratando de orientar de la mejor manera posible al usuario para que no tenga dificultades durante su uso.

También se han añadido botones de ayuda en los menús de selección de corpus y obras en la parte inferior derecha, que muestran una pequeña alerta explicando al usuario las funcionalidades de esa pantalla.

E.8 Wiki

Como la wiki del proyecto se encontraba desactualizada puesto que la que se referenciaba era la de la versión 1.0, se ha creado una nueva actualizando con todas las funcionalidades nuevas en el siguiente enlace:

<http://wikinetextractor-3-0.wikidot.com/>

Se ha realizado con WikiDot [43] y ha supuesto un esfuerzo importante, ya que el editor de texto no es el más sencillo de usar, pero se consideró un esfuerzo necesario, ya que mejora considerablemente la documentación y la ejecución de la aplicación.

Se ha añadido una breve introducción al proyecto y la guía de usuario, para explicar pantalla a pantalla todas las funcionalidades que se pueden encontrar. Para mantener el aspecto de la internacionalización realizada en el punto anterior, la wiki también se encuentra en inglés y español

E.9 Despliegue

El servidor elegido para alojar la aplicación es Railway, una plataforma que permite albergar en la nube aplicaciones web y diferentes despliegues en diferentes lenguajes de programación. Es una herramienta muy sencilla de usar, con un despliegue muy rápido e intuitivo y que nos ofrece quinientas horas mensuales de uso de la aplicación desplegada. .[48]

Durante el despliegue ha surgido un problema y es que, con la aplicación desplegada, la exportación a vídeo no funciona correctamente. Esto es debido a que, para hacer la exportación, hay que establecer una variable PATH que indica donde se encuentra el intérprete, indicando un fichero .exe que se encuentra en nuestro proyecto. Esa referencia en el despliegue da error, indicando que no se tienen los permisos necesarios para acceder a ese fichero. Se ha investigado y no se ha llegado a encontrar una solución a tiempo, puesto que el problema es que, para despliegue, la plataforma maneja una powershell de Linux, la cual impide abrir ficheros con la extensión exe por condiciones de seguridad, lo que obligaba a cambiar todo el método de exportación a

vídeo y buscar otro nuevo, ya que las alternativas manteniendo el formato ya fueron investigadas en su día y no daban resultado.

Se ha eliminado esta funcionalidad para el despliegue y se ha incluido para la entrega una máquina virtual con la aplicación lista para ejecutar en local, con el fin de demostrar que la exportación a vídeo funciona correctamente.

F. TRABAJOS RELACIONADOS

Dado que es uno de los campos que más desarrollo y estudio ha tenido en los últimos años, existen multitud de trabajos relacionados con la ciencia de redes. Sin embargo, con un enfoque más casual y no tan académico o científico no destacan tantos.

Algunos ejemplos de trabajos similares en cuanto a enfoque y desarrollo con el proyecto de NetExtractor van a ser comentados.

F.1 UbuBookNet

Fue el punto de partida de la primera versión de NetExtractor y sirvió como guía y ejemplo para su desarrollo, llegando a incorporar hasta en la versión 3.0 características que se desarrollaron en él, por tanto, tiene que ser la primera mención de este apartado.[52]

Este proyecto, realizado por Luis Miguel Cabrejas Arce en 2019, consiste en desarrollar una aplicación web que permite obtener, visualizar y analizar redes basadas en personajes de novelas que se introducen en formato ePub. Para obtener los personajes, se utiliza un analizador léxico que lee el ePub y considera como personajes aquellas palabras que aparecen en mayúscula con su primera letra. Sin embargo, esto puede generar muchos resultados incorrectos, por lo que la aplicación incluye opciones adicionales. Además, se puede obtener un diccionario de personajes a través de una wiki de fandom, o importar un archivo CSV que contenga los nombres de los personajes.

En este proyecto, las interacciones entre personajes se definen cuando dos personajes aparecen dentro de un rango de palabras. Este rango no es fijo, sino que el usuario puede ingresar el número máximo de palabras para considerar una interacción válida. Además, el usuario puede especificar el número mínimo de apariciones que debe tener un personaje, decidir si se deben tener en cuenta los capítulos para identificar los personajes y modificar el diccionario de personajes (agregar, eliminar o modificar identificadores).

Finalmente, el usuario puede visualizar la red de personajes utilizando la herramienta de visualización de redes Netwulf. Esta herramienta permite ajustar varios parámetros para mejorar la visualización. Además, el usuario puede ver las características seleccionadas en un informe final.

F.2 NetExtractor 1.0 y 2.0

Fueron las dos primeras versiones del proyecto y han servido como punto de partida, guía y ejemplo para varios de los desarrollos que se han llevado a cabo.

En el caso de la primera versión, desarrollada por Jorge Navarro en 2020 [51], se implementó a partir del proyecto de UbuBookNet y mejoró la extracción de personajes añadiendo la posibilidad analizar redes para películas pasadas como url, mediante la lectura de un guion de la página *imsdb*. Consideraba como personajes aquellos que estén dentro de una escena y aparezcan en negrita. En cuanto a las interacciones entre personajes, en el caso de los guiones de películas, se crea un enlace entre dos personajes si están presentes en la misma escena.

Mejoró considerablemente la visualización la red utilizando la herramienta "network styling with d3"[37], permitiendo ajustar parámetros visuales de la red, como el tamaño de los nodos, el grosor de los enlaces, la distancia entre nodos, entre otros.

Por último, incorporó mejoras y nuevas métricas al informe final, consiguiendo obtener un informe más preciso y concreto.

En el caso de la versión 2.0, se añadieron las redes dinámicas, permitiendo al usuario visualizar la evolución de la red en cada uno de sus instantes y poder ver todas las variaciones que iba sufriendo la red con el paso del tiempo. Se incorporaron diversas exportaciones para poder visualizar esta evolución en distintos formatos.

También se modificaron los informes, para poder obtener las diferentes métricas en función del instante seleccionado y su evolución previa hasta llegar a ese instante.

F.3 Network of Thrones

En este estudio [53], Andrew Beveridge y Jie Shan realizaron un análisis detallado del tercer libro de la saga "Juego de Tronos", llamado "Tormenta de Espadas". Su objetivo era transformar tanto la novela como la exitosa serie de televisión en una red de interacciones entre los diversos personajes que aparecen en la historia.

Como resultado de su trabajo, lograron construir una red compuesta por 107 nodos, que representan a los personajes principales de la novela, y un total de 353 enlaces que conectan estos nodos. Cada enlace tenía asignado un peso, el cual se incrementaba cada vez que dos personajes interactuaban entre sí, al igual que en NetExtractor.

Para definir una interacción, los autores establecieron 15 palabras en la narrativa, a modo de distancia máxima para que dos personajes tuvieran enlace entre ellos.

Posteriormente, se analizó la importancia de los personajes en la red utilizando diversas medidas de centralidad, las cuales son ampliamente reconocidas y utilizadas en este tipo de estudios. También se llevaron a cabo análisis de las comunidades presentes en la red utilizando el algoritmo de Louvain [34], el cual también ha sido empleado en el proyecto NetExtractor.

F.4 Evaluating named entity recognition tools for extracting social networks from novels

Los autores de este estudio, Niels Dekker, Tobias Kuhn y Marieke van Erp, se enfocaron en la extracción de características en redes complejas basadas en novelas. Para llevar a cabo este trabajo, seleccionaron cuarenta novelas, divididas en veinte clásicas y veinte modernas.

Con el fin de detectar los personajes, se eliminaron manualmente secciones no relevantes de las novelas, como agradecimientos, notas de la editorial o información del autor. De esta manera, se centró la detección de personajes exclusivamente en la trama. Además, se definió la interacción como la aparición de dos personajes en la misma frase.

De este modo, se crearon redes de personajes extrayendo las siguientes características:

- Grado medio de la red.
- Grado medio de la red considerando el peso de los enlaces.
- Longitud media del camino.
- Diámetro.
- Densidad.
- Modularidad.
- Componentes conectados.

- Coeficiente medio de agrupamiento.

Una vez que se extrajeron estas características, se realizó un estudio comparativo entre las medidas de las novelas modernas y las clásicas, analizando las diferencias. Se concluyó que las diferencias entre ambos grupos eran mínimas, siendo la densidad, la única característica destacada que diferenciaba en general a las novelas clásicas, las cuales tendían a ser más densas que las modernas.[54]

F.5 Character network analysis of Émile Zola's Les Rougon-Macquart

En este trabajo Yannick Rochat, empleó métodos de análisis de redes para esbozar una tipología de novelas de ficción a partir de los personajes y su proximidad en la narración. Generó las redes de personajes modelando las veinte novelas que componen Les Rougon-Macquart, una saga de novelas escritas por Émile Zola.

Para clasificarlos y construir la red, emplearon métodos que rastrean los caracteres principales y secundarios en relación con los sistemas de caracteres. Para el análisis de redes, utilizaron medidas de centralidad como el grado y la centralidad del vector propio, explicadas con anterioridad en los conceptos teóricos. Con este análisis de un pequeño corpus, consideraron la posibilidad de ampliar a un análisis a gran escala de novelas a través de sus redes de personajes.[55]

G. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

G.1 Conclusiones

Una vez concluido el trabajo, se podría decir que en su totalidad o casi, se han llegado a cumplirlos objetivos fijados a comienzo del curso con la elección de este trabajo.

A nivel de desarrollo se ha logrado incorporar las obras de teatro, sin tener que limitar apenas las funcionalidades ya existentes. Se han mejorado las funcionalidades dinámicas para una mejor visualización. Se han solucionado los errores que pudiesen aparecer, bien por la incorporación del nuevo módulo o por librerías desactualizadas.

La experiencia para el usuario también ha sido mejorada, incluyendo ayudas para la ejecución tanto en la aplicación como ayudas externas actualizando completamente la wiki, junto con la internacionalización del proyecto.

A nivel personal, bastante satisfecho con el trabajo realizado. Quizás el trabajo con la metodología SCRUM podría haberse realizado mejor si lo hubiese seguido de un modo más estricto y eso me hubiera llevado a realizar alguna implementación extra, pero al trabajar diariamente con esta metodología en mi trabajo, no era especialmente uno de mis objetivos más importantes.

Me hubiera gustado trabajar más con métricas de redes, ya que a parte de revisiones de las ya existentes no he llegado a realizar ningún desarrollo por mi cuenta, consecuencia probablemente de no haber seguido un ritmo de trabajo estricto.

G.2 Líneas de trabajo futuras

Se van a sugerir aspectos mejoras que se podrían realizar en un futuro de cara a una próxima versión de este trabajo.

Internacionalización

No porque sea precisamente necesaria, más bien por lo sencillo que puede resultar. En este trabajo porque se abordó tarde este tema y no era algo principal, pero con una guía bien explicada de como se hace es extremadamente sencillo de realizar y es algo que nunca viene mal.

Guiones de películas

Es una sugerencia al aire porque no es sencillo encontrar una alternativa, pero si se consiguiese algo parecido a Dracor para las películas sería una mejora increíble. La opción actual está bien, pero está desactualizada desde la versión 1.0 de este proyecto, así que incluir una alternativa a esto elevaría la calidad del proyecto.

Nuevas métricas

La incorporación de nuevas métricas es algo que siempre viene bien para un trabajo de este tipo. Quizás no las métricas más estudiadas y analizadas, sino métricas y análisis que se están realizando hoy en día, como puede ser el porcentaje de participación de las mujeres, estudiado en el “Test de Bechdel”[56]

Funcionalidades de Dracor

Dracor devuelve mucha información acerca de las obras. Estaría bien investigar a fondo todas las posibilidades, de cara a nuevas funcionalidades o métricas.

Detección etnia obras de teatro

Sabiendo que las obras de teatro están agrupadas en su mayoría por país de origen, estaría bien tratar de evitar el predictor y tratar de asignar una etnia a los personajes más acertadas, porque queda un poco raro elegir el corpus español y que la mayoría de los personajes no salgan como hispanos.

Eliminación de ruido en epub y películas

La captación de personajes en estos dos módulos suele llegar con cierto “ruido”, tomándose todo elemento que empieza por mayúsculas en los epub, cogiendo así accidentes geográficos, ciudades... o en el caso de las películas los números de escena y similares. Se podría emplear alguna herramienta tipo chatgpt o similar para que elimine todo aquello que consideremos innecesario.

BIBLIOGRAFÍA

- [1] la enciclopedia libre Wikipedia, «Teoría de grafos». https://es.wikipedia.org/w/index.php?title=Ciencia_de_redes&oldid=146505546
- [2] A.-L. Barabási, «Network Science Book», 5 de agosto de 2016. <http://networksciencebook.com/>
- [3] Proyectos Ágiles, «Scrum», [En línea]. Disponible en: <https://proyectosagiles.org/que-es-scrum/>
- [4] M. Newman, *Networks: An Introduction*. 2010. [En línea]. Disponible en: https://math.bme.hu/~gabor/oktatas/SztoM/Newman_Networks.pdf
- [5] R. J. Trudeau, *Introduction to Graph Theory*. 1993. [En línea]. Disponible en: https://books.google.gg/books?id=eRLEAgAAQBAJ&printsec=copyright&source=gbs_pub_info_r#v=onepage&q&f=false
- [6] D. Easley y J. Kleinberg, *Networks, Crowds, and Markets*. 2010. [En línea]. Disponible en: <https://www.cs.cornell.edu/home/kleinber/networks-book/>
- [7] M. O. Jackson, *Social and Economic Networks*. 2008.
- [8] la enciclopedia libre Wikipedia, «EPUB». <https://es.wikipedia.org/w/index.php?title=EPUB&oldid=149843511>
- [9] Academia Lab, «Distribución de grados». <https://academia-lab.com/enciclopedia/distribucion-de-gradoss/>
- [10] W. La Enciclopedia libre, «Coeficiente de agrupamiento», [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Coeficiente_de_agrupamiento&oldid=146241368
- [11] W. La Enciclopedia Libre, «Cliques». <https://es.wikipedia.org/w/index.php?title=Clique&oldid=144550168>
- [12] Robert A. Hanneman, «INTRODUCCIÓN A LOS MÉTODOS DEL ANÁLISIS DE REDES SOCIALES», *Revista Redes Rediris*, [En línea]. Disponible en: <http://revista-redes.rediris.es/webredes/textos/cap7.pdf>

- [13] W. La Enciclopedia Libre, «Degeneracy (graph theory)». [https://en.wikipedia.org/w/index.php?title=Degeneracy_\(graph_theory\)&oldid=1136395078](https://en.wikipedia.org/w/index.php?title=Degeneracy_(graph_theory)&oldid=1136395078)
- [14] M. Girvan y M. E. J. Newman, «Community structure in social and biological networks», *Proceedings of the National Academy of Sciences*, vol. 99, n.º 12, pp. 7821-7826, 2002, doi: 10.1073/pnas.122653799.
- [15] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, y E. Lefebvre, «Fast unfolding of communities in large networks», *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, n.º 10, p. P10008, oct. 2008, doi: 10.1088/1742-5468/2008/10/P10008.
- [16] A. Clauset, M. E. J. Newman, y C. Moore, «Finding community structure in very large networks», ago. 2004, doi: 10.1103/PhysRevE.70.066111.
- [17] G. Palla, I. Derenyi, I. Farkas, y T. Vicsek, «Uncovering the overlapping community structure of complex networks in nature and society», *Nature*, vol. 435, n.º 7043, pp. 814-818, jun. 2005, [En línea]. Disponible en: <http://dx.doi.org/10.1038/nature03607>
- [18] R. Guimerà y L. Amaral, «Functional Cartography of Complex Metabolic Networks», *Nature*, vol. 23, pp. 22-231, jun. 2005.
- [19] W. La Enciclopedia Libre, «Web Scraping». https://es.wikipedia.org/w/index.php?title=Web_scraping&oldid=147223363
- [20] E. Equipo Editorial, «Obra de teatro». <https://humanidades.com/obra-de-teatro/>
- [21] K. Schwaber y J. Sutherland, «Scrum Guides ». <https://scrumguides.org/>
- [22] «Github». <https://github.com/>
- [23] «ZenHub». <https://www.zenhub.com/>
- [24] «Visual Studio Code». [En línea]. Disponible en: <https://code.visualstudio.com/>
- [25] «Mendeley». <https://www.mendeley.com/>
- [26] «Flask Guide». [En línea]. Disponible en: <https://flask.palletsprojects.com/en/2.3.x/>
- [27] M. Otto y J. Thornton, «getBootstrap». [En línea]. Disponible en: <https://getbootstrap.com/>

- [28] W. The Free Encyclopedia, «Bootstrap». [https://en.wikipedia.org/w/index.php?title=Bootstrap_\(front-end_framework\)&oldid=1157109332](https://en.wikipedia.org/w/index.php?title=Bootstrap_(front-end_framework)&oldid=1157109332)
- [29] Mozilla, «JavaScript». [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [30] «jQuery». [En línea]. Disponible en: <https://jquery.com/>
- [31] «DataTables». [En línea]. Disponible en: <https://datatables.net/>
- [32] F. Fischer, «Programmable Corpora: Introducing DraCor, an Infrastructure for the Research on European Drama», Zenodo, jul. 2019. doi: 10.5281/zenodo.4284002.
- [33] V. Torvik y S. Agarwal, «Ethnea –an instance-based ethnicity classifier based on geo-coded author names in a large-scale bibliographic database», jun. 2016.
- [34] Aynaud T. taynaud, «Python-Louvain». [En línea]. Disponible en: <https://github.com/taynaud/python-louvain>
- [35] «NetworkX». [En línea]. Disponible en: <https://networkx.org/>
- [36] G. Rosetti, «DyNetX». [En línea]. Disponible en: <https://dynetx.readthedocs.io/en/latest/>
- [37] M. B. Aslak U, «Network styling with D3».
- [38] Chunquisi, «Gexfpy». [En línea]. Disponible en: <https://github.com/chunqishi/gexfpy>
- [39] «Gephi». [En línea]. Disponible en: <https://gephi.org/>
- [40] «Matplotlib». [En línea]. Disponible en: <https://matplotlib.org/>
- [41] «Animation de Matplotlib». [En línea]. Disponible en: https://matplotlib.org/stable/api/animation_api.html
- [42] «FFmpeg». [En línea]. Disponible en: <https://ffmpeg.org/>
- [43] «WikiDot». <https://www.wikidot.com/>
- [44] «Poedit». [En línea]. Disponible en: <https://poedit.net/>
- [45] «Flask Babel». [En línea]. Disponible en: <https://python-babel.github.io/flask-babel/>

- [46] «Draw.io». <https://app.diagrams.net/>
- [47] «EasyRetro.io». <https://easyretro.io/>
- [48] «Railway». [En línea]. Disponible en: <https://railway.app/>
- [49] F. Fischer y E. Grigoryev, «USING WEBWEB AND NETWULF WITH DRACOR'S API». <https://weltliteratur.net/netwulf-webweb/>
- [50] Gephi, « User Manual, Import Dynamic Data». https://docs.gephi.org/User_Manual/Import_Dynamic_Data/
- [51] J. Navarro, «NetExtractor 1.0», 2020. <https://github.com/jorgenavarroglez/NetExtractor>
- [52] L. M. Cabrejas Arce, «UbuBookNet», 2019. <https://github.com/lca0037/GII18.0U-Ububooknet>
- [53] A. Beveridge y J. Shan, «Network of Thrones», *Math Horizons*, vol. 23, p. 18, jun. 2016, doi: 10.4169/mathhorizons.23.4.18.
- [54] N. Dekker, T. Kuhn, y M. van Erp, «Evaluating named entity recognition tools for extracting social networks from novels». [En línea]. Disponible en: <https://peerj.com/articles/cs-189/>
- [55] Y. Rochat, «Character network analysis of Émile Zola's Les Rougon-Macquart», 2015. [En línea]. Disponible en: <https://infoscience.epfl.ch/record/210573>
- [56] W. La enciclopedia libre, «Test de Bechdel». https://es.wikipedia.org/w/index.php?title=Test_de_Bechdel&oldid=151156191