UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Gº en Ingeniería Informática

TFG Ingeniería Informática:

NetExtractor

Presentado por Jorge Navarro González
en Burgos el 13 febrero de 2020
Tutores D. José Manuel Galán Ordax
y Dña. Virginia Ahedo García

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

Exponen:

Que el alumno D. Jorge Navarro González, con DNI 13173430L, ha realizado el TFG en Ingeniería Informática titulado: *NetExtractor*.

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

En Burgos a 13 de febrero de 2020

Vº. Bº del Tutor

Vº. Bº. del Tutor

Resumen

En la era de la complejidad y el Big Data, la ciencia de redes es uno de los campos de investigación que goza de mayor popularidad, ya que proporciona un amplio rango de herramientas analíticas y computacionales que permiten modelar, analizar y entender los sistemas interconectados.

Sin embargo, muchas de las herramientas existentes para análisis de redes pueden resultar complejas de utilizar por requerir de conocimientos de programación y/o de un dominio previo del tema.

Por ello, este trabajo surge con la intención de proporcionar una herramienta de análisis de redes para todos los públicos, que permita a cualquier persona interesada en el análisis de las redes complejas, familiarizarse con ellas y calcular las métricas más comunes de una forma interactiva y sin necesidad de programar.

El enfoque escogido, a su vez, no puede ser más divulgativo: nuestra aplicación web está diseñada para obtener las redes de interacción entre personajes de novelas y/o guiones de películas. De este modo, el usuario podrá personalizar su aprendizaje de teoría de redes escogiendo la película o novela a analizar en base a sus gustos, lo que sin duda aumentará su interés por la disciplina y le facilitará la comprensión de los distintos conceptos.

En concordancia con todo lo anterior, NetExtractor, ha sido desarrollada en forma de aplicación web a la que se puede acceder a través del siguiente enlace: https://netextractor.herokuapp.com/.

Descriptores

Generador de redes de interacción, visualización de la red, informes sobre la red, aplicación web, Python, análisis de comunidades, análisis de roles, predictor de género y etni

Abstract

In the era of complexity and Big Data, network science is one of the most popular research fields, since it provides a wide range of analytical and computational tools that allow modeling, analyzing and understanding interconnected systems.

However, many of the existing tools for network analysis can be complex to use because they require programming knowledge and/or a previous domain of the subject.

Therefore, this work arises with the intention of providing a network analysis tool for all audiences, which allows anyone interested in the analysis of complex networks calculate the most common metrics in an interactive way and without programming.

The chosen approach, in turn, cannot be more outreaching: our web application is designed to obtain interaction networks between characters from novels and/or movie scripts. In this way, users can customize their network-theory learning by choosing the movie or novel to analyze based on their interests, which will undoubtedly increase their interest in the discipline and help them understand the different concepts.

In accordance with all the above, NetExtractor has been developed in the form of a web application that can be accessed through the following link: https://netextractor.herokuapp.com.

Keywords

Interaction network generator, network displayer, network reports, web application, Python, community analysis, role analysis, sex and ethnicity prediction.

Índice General

Índice de Ilustraciones

A. Introducción

La ciencia de redes es una disciplina que se dedica al estudio de las redes complejas y de sus características. Una red está constituida por un conjunto de elementos llamados nodos y las interacciones entre ellos, denominadas enlaces. Existen diversos tipos de redes que se pueden analizar, como pueden ser redes biológicas, redes de telecomunicaciones, redes informáticas, Quizá uno de los aspectos más relevantes de la teoría de redes es que con independencia de lo distintos que sean en realidad los sistemas que se modelan mediante redes, todas ellas pueden analizarse mediante un conjunto de herramientas matemáticas y computacionales común (1)(2).

Actualmente las redes están presentes en todos los ámbitos de nuestra vida, desde que cogemos nuestro móvil y nos creamos un perfil en Twitter, ya estamos siendo parte de una red. En medicina las redes tienen una gran utilidad, desde el análisis de distintos grupos dentro de la sociedad con el fin de poder prevenir propagaciones de enfermedades, hasta la utilización de medidas de centralidad como la de valor propio(3) para extraer información del cerebro durante una resonancia magnética. En transporte, se extraen características de la red de carreteras para poder obtener los caminos más cortos y rápidos, o para conseguir los recorridos óptimos para lograr el máximo alcance con el menor recorrido cuando se diseñan las líneas de autobús. En redes sociales, es común realizar análisis de comunidades para identificar a los distintos grupos sociales y caracterizarlos. En el ámbito empresarial, las redes se emplean por ejemplo para identificar qué tipo de mercado se adapta mejor a las características de tu empresa, y poder así aumentar la rentabilidad, o para estudiar las élites empresariales y cómo están conectadas(4). Cuando accedemos a nuestro navegador de Google, se están usando redes con el fin de dirigir tus búsquedas a las páginas que tengan mayor importancia con el algoritmo de PageRank(5). Por todo esto, no cabe duda de que las redes tienen cada vez más peso en nuestra sociedad, y de que, por consiguiente, se hace necesario democratizar su conocimiento entre el público general.

Desde un punto de vista educativo, es importante destacar el potencial pedagógico de las redes y lo bien que se adaptan a metodologías docentes de diferenciación por interés, consistentes en permitir al alumno aplicar los conocimientos adquiridos en clase a temas relacionados con sus aficiones e intereses personales.

Muchas de las herramientas usadas hoy en día para realizar análisis de redes y para la enseñanza de estas, su enseñanza necesita requieren de conocimientos de programación, de un dominio previo de la teoría de redes o de ambas, no resultando adecuadas para el público general. Por todo ello, en este proyecto presentamos una aplicación web que permite obtener de manera semiautomática las redes de interacción entre los personajes de novelas y/o películas. Dichas redes podrán usarse como ejemplos ilustrativos en clase o en prácticas para que el alumno trabaje sobre ellas.

Este proyecto además parte de uno anterior, Ububooknet, en el cual se permitía al usuario extraer algunas de las características más importantes de redes complejas generadas a partir de los personajes de novelas.

NetExtractor presenta una visión más completa, añadiendo funcionalidad para la creación de redes a partir de películas, creando informes más completos, con un análisis minucioso de las comunidades y roles que aparecen en redes complejas implementando distintos algoritmos conocidos para ello, y que mejoran el coste de tiempo que supone su obtención; y añadiendo características a los personajes como la etnia y el sexo que se establecerán como atributos propios del nodo con el fin de poder obtener más información acerca de cada personaje a la hora de ser estudiado.

1.1 Estructura de la memoria

La memoria va a tener la siguiente estructura:

- **Introducción:** Descripción del proyecto, distribución de la memoria y anexos y enlaces a repositorios y aplicación.
- **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 cada uno de los conceptos considerados importantes para entender el proyecto.
- **Técnicas y herramientas:** Aspectos técnicos: Listado de las herramientas empleadas en el proyecto
- **Aspectos relevantes del desarrollo:** Aspectos de alta relevancia ocurridos durante la realización del proyecto
- **Trabajos relacionados:** Trabajos previos sobre el tema que trata el proyecto.
- **Conclusiones y líneas de trabajo futuras:** Conclusiones obtenidas una vez realizado el proyecto y posibles ideas de mejora/líneas de trabajo futuras que permitirían mejorar el proyecto actual (funcionalidades adicionales, aspectos técnicos a optimizar, 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:** Descripción de 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 navegación a lo largo de la aplicación.

1.2 Enlaces adicionales

Los enlaces adicionales del proyecto que podemos encontrar son:

- **Página web de la aplicación:**
  - https://netextractor.herokuapp.com.
- **Wiki de la aplicación:**
  - https://wikinetextractor.wikidot.com.
- **Repositorio:**
  - https://github.com/jng0020/NetExtractor.
- **Youtube:**
  - Ejecución simple: https://youtu.be/-4pfwa-tp_s.
  - Ejecución detallada: https://youtu.be/WL-U85159nM.

B. Objetivos del proyecto

En el siguiente apartado se van a describir los objetivos del proyecto realizado, tanto los de carácter general y técnico como los de carácter personal.

Este proyecto se basa en un trabajo previo titulado Ububooknet, de Luis Miguel Cabrejas Arce, el cual fue defendido el 09 de julio de 2019. Ububooknet se caracterizaba por la capacidad de extraer personajes de novelas en formato ePub mediante un analizador léxico, calcular sus apariciones en la novela y de esta forma crear una red de interacciones entre personajes en la cual los nodos son los personajes obtenido y los enlaces se corresponden con las interacciones que tienen entre los mismos.

2.1 Objetivos generales

En este apartado se van a describir los objetivos generales del proyecto, es decir, aquellos objetivos definidos al inicio del proyecto.

- Ampliar la aplicación web para permitir introducir guiones de películas o capítulos de series en formato html obtenidos de la página de https://www.imsdb.com/.
- Ampliar la funcionalidad en lo referente a las métricas de análisis de redes implementadas. Añadiremos la predicción de género, etnia, el análisis de roles y el análisis de comunidades mediante otros algoritmos:
  - Louvain(6).
  - Clauset-Newman-Moore(7).
  - Método de percolación(Comunidades y roles K-clique)(8).
- Mejorar la generación de informes y la visualización de los resultados de estos.
- Incorporar la funcionalidad de predicción de género y etnia desarrollada por Vetle I. Torvik and Sneha Agarwal en la librería Ethnea + Genni(9). De este modo, podremos caracterizar más detalladamente a los personajes (nodos). Se dará también la posibilidad de introducir dichos atributos manualmente (si no se desea realizar la obtención de forma automática con Ethnea + Genni).
- Mejorar la interfaz general de la aplicación de forma que sea más vistosa y atractiva al usuario.
- Adición de una wiki a modo de guía de usuario, que podrá ser consultada en cualquier momento durante el uso de la aplicación para resolver dudas.
- Solucionar un problema previo que tenía Ububooknet con las sesiones de usuario, el cual consistía en que, en ocasiones, al pulsar los botones de navegación de la aplicación, se cerraba la sesión y se salía de la aplicación.

2.2 Objetivos técnicos

En este apartado se definirán los objetivos de carácter técnico, es decir, los detalles de la implementación y las herramientas a utilizar.

- El desarrollo del proyecto se realizará conforme a la metodología Scrum(10) estudiada en 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 GitKraken para gestionar el repositorio local y remoto.

- Utilizar Python como lenguaje de programación base. En Python se realizará tanto la generación de redes de interacción como su posterior análisis. Para el análisis de redes propiamente dicho utilizaremos la librería NetworkX. Otras librerías de Python a utilizar serán: NumPy, Matplotlib, Python-louvain, Scipy, BeautifulSoup4 y Ply.
- Utilizar Flask para el desarrollo web.
- Utilizar el patrón de diseño Modelo-Vista-Presentador
- Utilizar MediaWiki y Wikidot para la realización de la wiki.
- Hacer web *scraping* para obtener los personajes que aparecen en la película/serie bajo consideración, trabajando sobre el repositorio de guiones disponible en la página https://www.imsdb.com/.
- Utilizar Ethnea y Genni(9) con el fin de predecir el sexo y etnia de los personajes, y así poder caracterizar mejor los nodos.
- Realizar tests que garanticen la calidad del código.
- Hacer disponible la aplicación y la wiki a través de la web.

2.3 Objetivos personales

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

- Aplicar la metodología SCRUM en proyecto real con el fin de habituarme a la misma y de comprender mejor su utilidad y funcionamiento.
- Aprender a desarrollar una aplicación web en Python utilizando Flask.
- Aprender a utilizar nuevas herramientas que no se conocían hasta el momento.
- Poner en práctica todos los conocimientos adquiridos sobre Ciencia de Redes.

C. Conceptos teóricos

En este apartado se van a definir una serie de conceptos teóricos necesarios para una mejor comprensión del proyecto.

3.1 EPUB

EPUB es un tipo de formato de libros electrónicos para leer textos e imágenes; a partir de EPUB3 se ha hecho posible también la lectura de audios. El formato se diseñó de forma redimensionable, para poder adaptarse a distintos tipos de pantallas y letras. Fue un formato creado por el Foro Internacional de Publicaciones Digitales (IDPF) como un formato específico para la visualización de libros (11).

Todo el contenido de los EPUB está almacenado en un fichero que contiene la extensión ".epub" y que es un archivo basado en ZIP. La diferencia es que el *Container Format* proporciona una forma de comprobar que el contenido de ese ZIP es un libro en formato EPUB.

Uno de los ficheros más importantes dentro de la estructura de un EPUB, es el que se encuentra en la carpeta "META-INF" llamado "container.xml", este fichero es el que va a dirigir el sistema de lectura a la raíz de la publicación, es decir, el Package Document. Este va a ser el documento que nos va a indicar el contenido de todos los documentos que se incluyen en la publicación y sus respectivos recursos, este documento viene con la extensión ".opf".

Existe en este fichero una etiqueta llamada "spine", esta etiqueta es la que va a indicar el orden en el que se van a ir sucediendo los distintos archivos a la hora de leer el EPUB.

3.2 Redes y redes complejas

En este apartado se definirá una parte básica de este proyecto, las redes, y en concreto las redes complejas, pero antes de entrar en el concepto de red compleja, debemos abordar primero el tema de red, es decir, qué es una red.

Una red va a ser un conjunto de nodos y los enlaces existentes entre dichos nodos, que representan una información que puede ser de diversos tipos, es decir, no todos los enlaces y nodos representan la misma información, o tienen la misma relevancia dentro de la red. Las redes van a tener un tamaño que dependerá del número de nodos que posea la red.

En cuanto a los nodos, una red puede contener nodos de una sola clase, en cuyo caso se denominan redes unimodales, , un ejemplo arquetípico de red unimodal es aquel en el que los nodos representan a personas; también podemos tener nodos de dos clases, en este caso se llaman redes bimodales; un ejemplo de red bimodal podría ser aquel en el que un nodo representa un laboratorio y otro nodo representa un financiador para un determinado laboratorio; por último, existen redes en las que los nodos pueden tener multitud de clases, estas son las llamadas redes multimodales; en ellas, por ejemplo, una clase puede representar personas, otra animales y otra objetos.

Dependiendo de los enlaces, también distinguimos entre dos tipos de redes: dirigidas y no dirigidas. Las redes dirigidas son aquellas en que la relación entre los nodos tiene una dirección y un sentido, es decir, va de un nodo origen a un nodo destino; un ejemplo de red dirigida podría ser la relación de padre-hijo; también puede darse el caso de que un enlace dirigido tenga el mismo origen y destino, en cuyo caso se denomina "auto-enlace". En cuanto a las redes no dirigidas, son aquellas en que un enlace únicamente indica la existencia de una relación entre los nodos., no

teniendo sentido el especificar sentido; un ejemplo de red no dirigida sería la red de contactos sexuales, en la que simplemente se especifica la relación que ha habido entre dos personas.

Atendiendo al número de enlaces por cada par de nodos podemos distinguir entre redes simples o múltiples. Las redes simples son aquellas en las que, por cada par de nodos, solo existe un enlace como máximo. En el caso de que haya más de un enlace por cada par de nodos la red pasaría a ser una red múltiple. También pueden clasificarse las redes dependiendo de la representación que tenga el enlace en la red, es decir, si un enlace representa simplemente la relación entre dos nodos, la red va a llamarse red binaria, pero se puede dar el caso en el que el enlace lleva asociado un peso; siguiendo con el ejemplo de los contactos sexuales, un enlace entre dos personas podría llevar un peso específico dependiendo de las veces que han mantenido relaciones los dos nodos, en este caso la red pasaría a ser una red pesada.

Con todas estas características de una red, se puede decir que una red compleja será aquella red que este formada por una cantidad alta de nodos que se conectan entre sí aportando cierta información relevante(12).

En este proyecto, las redes con las que trabajaremos serán las redes de interacción entre los personajes de novelas y/o películas; en ellas, los nodos representarán a los personajes, siendo por tanto todos de una misma clase: personajes. A su vez, los enlaces entre nodos van a ser pesados, es decir, van a tener un peso que cuantificará la intensidad de la relación, i.e., el número de veces que han coincidido en la novela o película.

Del mismo modo, cabe destacar que, en nuestras redes de interacción entre personajes, las interacciones son bidireccionales (aparecer juntos en una escena o en un mismo párrafo), por lo que los enlaces serán no dirigidos. Por todo esto, podemos concluir que las redes del proyecto

Figura 1 Red Compleja

van a ser redes simples (ya que cada par de nodos tiene como máximo un enlace), no dirigidas y pesadas (los enlaces tienen un peso que cuantifica la intensidad de la relación).

3.3 Grado de los nodos

El grado de los nodos (*k*) es un número que viene dado dependiendo de los enlaces asociados a dicho nodo. Existen dos formas en las que se puede calcular el grado dependiendo de si tenemos una red pesada o tenemos una red binaria. En caso de una red binaria el grado va a venir dado únicamente por el número de enlaces asociados a un nodo. Por el contrario, si tenemos una red pesada, el número de enlaces será el sumatorio del peso de cada uno de los enlaces asociados a dicho nodo.

También habrá que tener en cuenta para esta medida si la red es dirigida o no, en caso de no serlo, hay que tomar la medida explicada en el párrafo anterior para todos los enlaces asociados a dicho nodo; por el contrario, si la red es dirigida, se suele hacer una distinción entre grado de entrada y grado de salida, cogiendo simplemente los enlaces que entran o los enlaces que salen para calcular los grados de entrada y salida respectivamente.

Grado Medio

Se puede calcular el grado medio de una red mediante una fórmula muy sencilla, aunque se calcula de forma distinta dependiendo de si el grado medio es de una red dirigida o no dirigida (13):

- Red dirigida:
  - Para calcular el grado medio de una red dirigida debemos tener en cuenta la siguiente fórmula:
    - $\langle k \rangle = \frac{L}{N}$
- Red no dirigida:
  - Para calcular el grado medio de una red no dirigida debemos tener en cuenta la siguiente fórmula:
    - $\langle k \rangle = \frac{2L}{N}$

Para el entendimiento de estas fórmulas debemos tener en cuenta que:

- L: Número de enlaces de la red.
- N: Número de nodos de la red.

Distribución de grado

Una vez obtenido el grado de los nodos, se puede obtener otra característica de la red derivada del mismo, la distribución de grado (P(k)), que se define como la cantidad de nodos de la red con un cierto grado *k*, o dicho de otro modo, que indica la probabilidad de que un nodo seleccionado al azar tenga grado *k* (14).

La distribución de grado se calcula de la siguiente forma:

$$P(k) = {}^{N_K}/_{N}$$

Donde debemos tener en cuenta que:

- N: Número de nodos de la red.
- N_k: Número de nodos que tienen grado k.

Densidad

La densidad de una red viene dada por el número de enlaces que tiene la red y el número posible de enlaces que puede haber en la red. Con esta medida podemos obtener otra clasificación de redes en redes muy densas y redes poco densas(15).

Las redes muy densas van a ser las que tienen un número de enlaces mucho mayor que el número de nodos (L>>N). En caso contrario, es decir, cuando tengamos un número de enlaces similar al número de nodos, la red va a ser considerada una red poco densa.

3.4 Medidas de distancia

Dentro de las medidas de distancia de una red debemos hacer hincapié en el concepto de distancia geodésica; en una red binaria, la distancia geodésica entre dos nodos es la distancia mínima (medida en número de enlaces) entre el nodo origen y el nodo destino; en el caso de la red pesada, la distancia geodésica se va a calcular como la suma mínima de los pesos de los enlaces que hay que atravesar para ir de un nodo origen a un nodo destino.

Componentes conectados

Se define componente conectado como el conjunto de nodos entre los cuales siempre podremos desplazarnos de un nodo origen a un nodo destino a través de los enlaces que existen entre los nodos.

Excentricidad

La excentricidad de un nodo se puede definir como la distancia máxima que va a existir entre ese nodo y cualquier otro nodo de la red, siguiendo caminos de distancia mínima(16).

Diámetro

El diámetro de una red va a ser la longitud del camino más corto que une a los dos nodos más alejados de la red.

Otra forma de definir este concepto teniendo en cuenta la excentricidad sería: “el diámetro es la máxima de todas las excentricidades de la red”.

Al diámetro se lo denota con la letra D(16).

Radio

Así como el diámetro se definía como la máxima de las excentricidades, el radio se va a definir como la mínima de las excentricidades. En redes, dos veces el radio siempre va a ser igual o más grande que el diámetro.

3.5 Medidas de *clustering*

En este apartado nos vamos a centrar en hablar del coeficiente de *clustering* y de la transitividad.

Coeficiente de *clustering* local

El coeficiente de *clustering* local de una red mide la cantidad de vecinos de un nodo que están conectados entre sí. El coeficiente de clustering es de utilidad para calcular algunas medidas de centralidad de la red que serán explicadas posteriormente(17).

Para calcular esta medida en una red no dirigida se emplea la siguiente fórmula:

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

Donde debemos tener en cuenta que:

- C_i: Coeficiente de clustering local del nodo i.
- k_i: Grado del nodo i.
- L_i: Número de enlaces que tiene el nodo i.

Para calcular esta medida en una red dirigida bastaría con eliminar el "2" en el numerador.

Coeficiente de *clustering* local medio

Una medida global del coeficiente de *clustering* es el coeficiente de *clustering* local medio con el cual se calcula una media de todos los coeficientes de *clustering* locales de cada nodo(17).

Para calcular el coeficiente de *clustering* local medio se emplea la siguiente fórmula:

$$\langle C \rangle = \frac{1}{N}\sum_{i=1}^{N} C_i$$

Donde debemos tener en cuenta que:

- N: Número de nodos de la red.
- <C>: Coeficiente de *clustering* local medio.
- C_i: Coeficiente de *clustering* local del nodo i.

Transitividad

Aunque en ocasiones se emplea el mismo nombre para llamar al coeficiente de *clustering* local que a la transitividad, se calculan de una forma distinta. La transitividad mide el número de tríadas abiertas, las tríadas son los triángulos que se forman en la red entre tres nodos y se calcula de la siguiente forma:

$$Transitividad = \frac{3 \times Nº\ de\ tríadas\ cerradas}{Nº\ de\ tríadas\ abiertas}$$

3.6 Medidas de centralidad

Las medidas de centralidad nos sirven para determinar qué nodos son los más importantes de la red atendiendo a distintos criterios; por ello, el nodo más importante según una de ellas no

tiene por qué ser el nodo más importante de acuerdo con otra de las métricas de centralidad. En todos los gráficos que van a aparecer para cada una de las medidas de centralidad descritas, los nodos tienen un tamaño proporcional al valor que poseen de la métrica en cuestión, pudiendo ser clasificados visualmente de más importante a menos importante según su tamaño. Todas las imágenes han sido obtenidas de la aplicación utilizando la librería de Python NetworkX.

Centralidad de grado

La centralidad de grado puede ser definida como el número de enlaces que inciden sobre un nodo. Si existe una red dirigida, debemos tener en cuenta que dentro de la centralidad de grado existen dos medidas, la centralidad de grado *in-degree*, de los enlaces que apuntan al nodo, y la centralidad de grado *out-degree*, de los enlaces que salen del nodo y apuntan a otro (18).

Esta medida tiene algunas desventajas, sólo tiene en cuenta el grado que tiene un nodo y no si ese nodo está relacionado con nodos importantes de la red, lo que significa que, si te apunta un nodo poco relevante en la red, tiene el mismo valor que si te apuntara el nodo más importante.

Esta medida tiene algunas desventajas, sólo tiene en cuenta el grado que tiene un nodo y no si ese nodo está relacionado con nodos importantes de la red, lo que significa que, si te apunta un nodo poco relevante en la red, tiene el mismo valor que si te apuntara el nodo más importante.

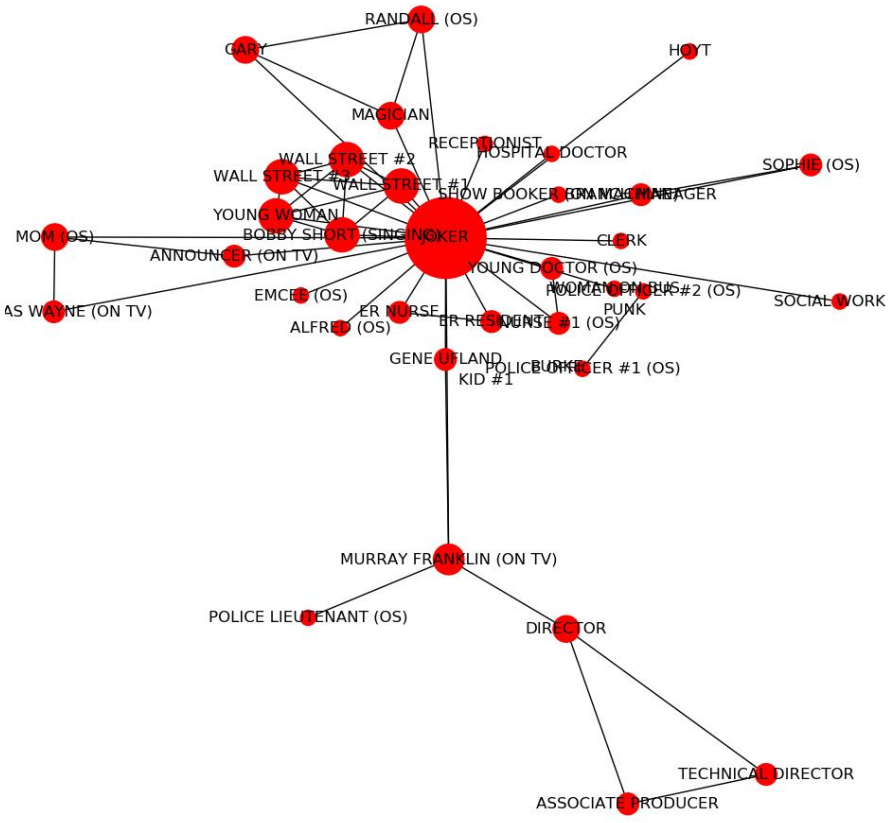

Figura 2 Centralidad de grado

Centralidad de cercanía

La centralidad de cercanía nos sirve para comprobar cómo de cerca está un nodo respecto al resto de nodos de la red (19).

De esta forma, de acuerdo a esta métrica vamos a considerar más importantes a los nodos que estén más cerca de otros nodos; cuanto más cerca estés del resto de nodos de la red, mayor será tu importancia de cercanía o *closeness centrality*.

Esta medida tiene factores que hacen que no sea muy adecuada en determinadas circunstancias. Por ejemplo, en redes muy pequeñas, en las que para llegar de un nodo a otro se necesitan muy pocos pasos, esta medida nos dará un valor muy parecido para todos los nodos que componen la red, no permitiéndonos discriminar entre unos y otros.

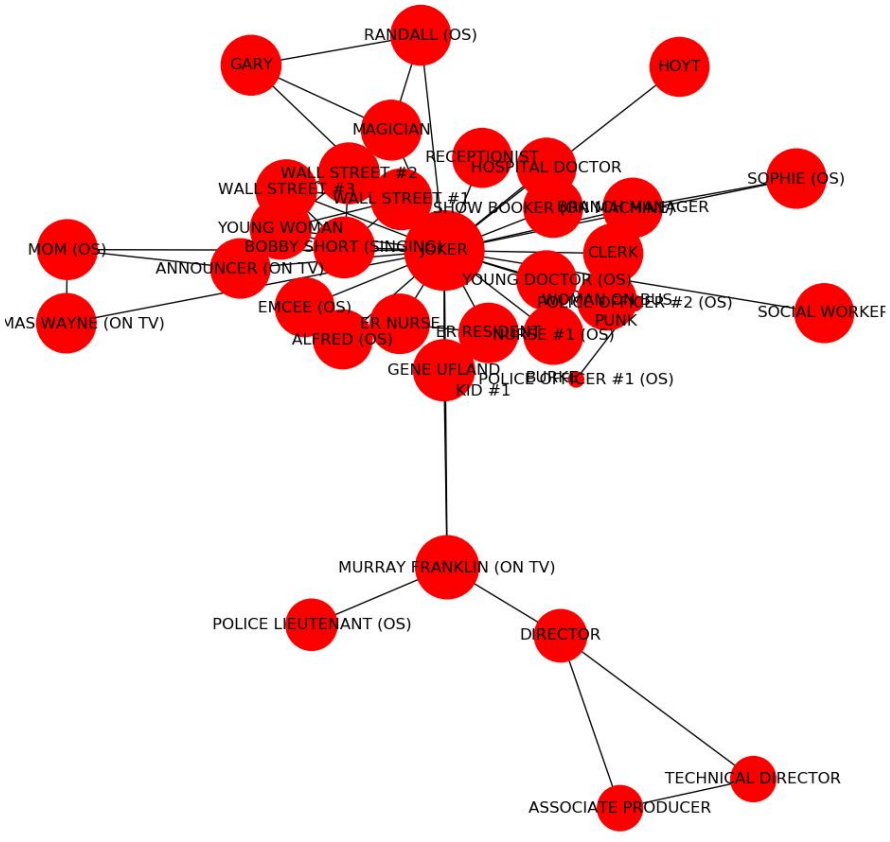

Figura 3 Centralidad de cercanía

En la imagen de la figura 3 podemos ver una red pequeña en la que el tamaño de los nodos es proporcional a su valor de centralidad de cercanía. En este ejemplo vemos claramente cómo todos los nodos tienen un valor de centralidad de cercanía muy parecido, por lo que esta métrica no nos permite discriminar entre unos y otros.

Centralidad de intermediación

La centralidad de intermediación se centra en los caminos más cortos entre dos nodos “x” e “y” que pasan por un nodo “n”. Es decir, la medida de centralidad de intermediación hace que los nodos más importantes sean aquellos que unen otros nodos. Por ejemplo, en dos grupos de amigos distintos, si hay un amigo común a los dos grupos, ese nodo sería un nodo muy importante según la medida de centralidad de intermediación ya que es el nodo que uno dos grupos enteros (20).

El problema que tiene esta medida es el alto coste computacional que supone implementarla y calcularla.

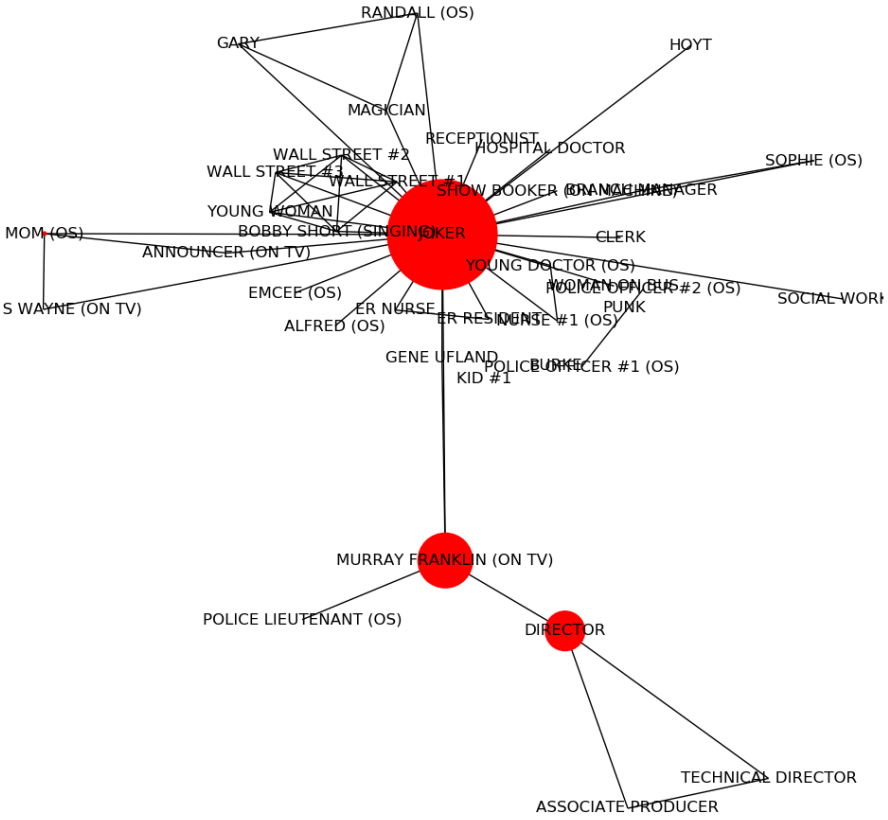

Figura 4 Centralidad de intermediación

Intermediación de camino aleatorio

La medida de intermediación de camino aleatorio es una media similar a la centralidad de intermediación, la diferencia reside en que, para calcularla, no se cogen los caminos más cortos entre dos nodos que pasan por un nodo en concreto, sino que se cogen caminos de forma aleatoria, no necesariamente los más cortos, que pasan por un nodo en concreto (21).

Centralidad de valor propio

La centralidad de valor propio es una medida de centralidad que hace que las conexiones entre nodos puedan tener un valor variable. Esta centralidad considera que la importancia de un nodo no solo depende del número de conexiones que posee con otros nodos, sino también de con quién está conectado (estar conectado con nodos importantes te hace más importante). De esta forma, un nodo puede tener un alto valor de centralidad de valor propio con unos pocos enlaces. En la centralidad de valor propio, cuanto más importantes sean los nodos a los que te conectas, más importante eres.

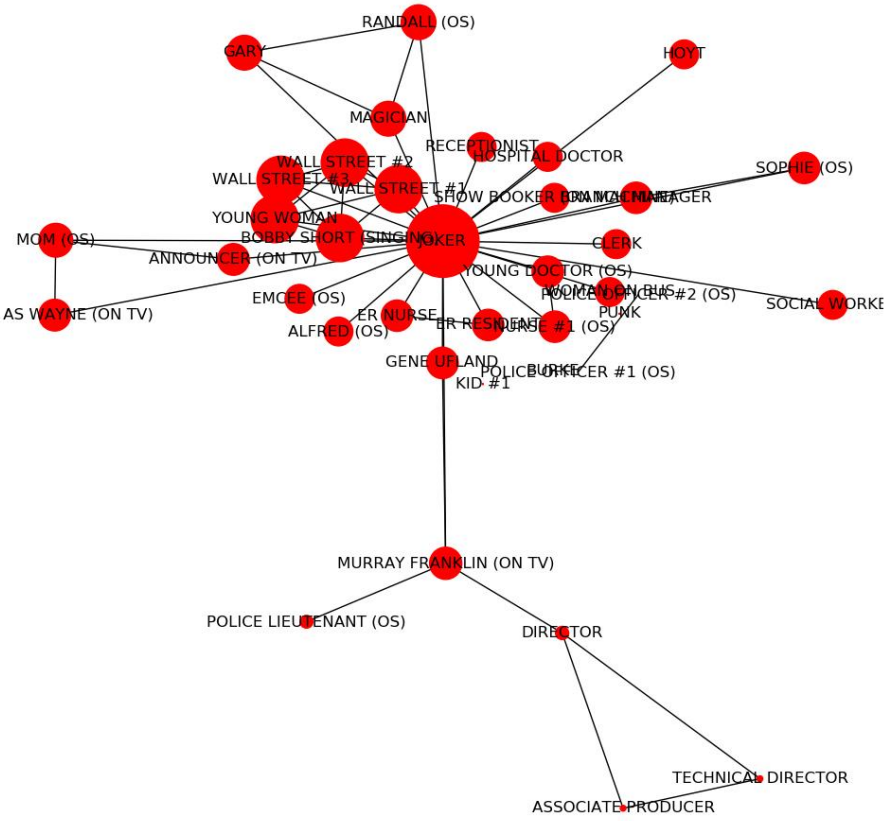

Figura 5 Centralidad de valor propio

PageRank

PageRank(5) es una medida de centralidad que se crea a partir de la centralidad de valor propio. Como medida de centralidad aparte, incluye una serie de cambios:

- La importancia que recibe un nodo de sus vecinos va a ser proporcional a su centralidad entre el número de nodos a los que aporta importancia.
- Parámetro beta: es una constante positiva para evitar que haya valores nulos en el cálculo.
- Parámetro alfa: sirve para modular la importancia con respecto a la centralidad de un nodo.

Google utiliza este algoritmo para sus sistemas de recomendación de páginas, una página va a ser muy importante dependiendo de los enlaces que apuntan hacia ella, es decir, dependiendo de cuantas páginas la apuntan (5).

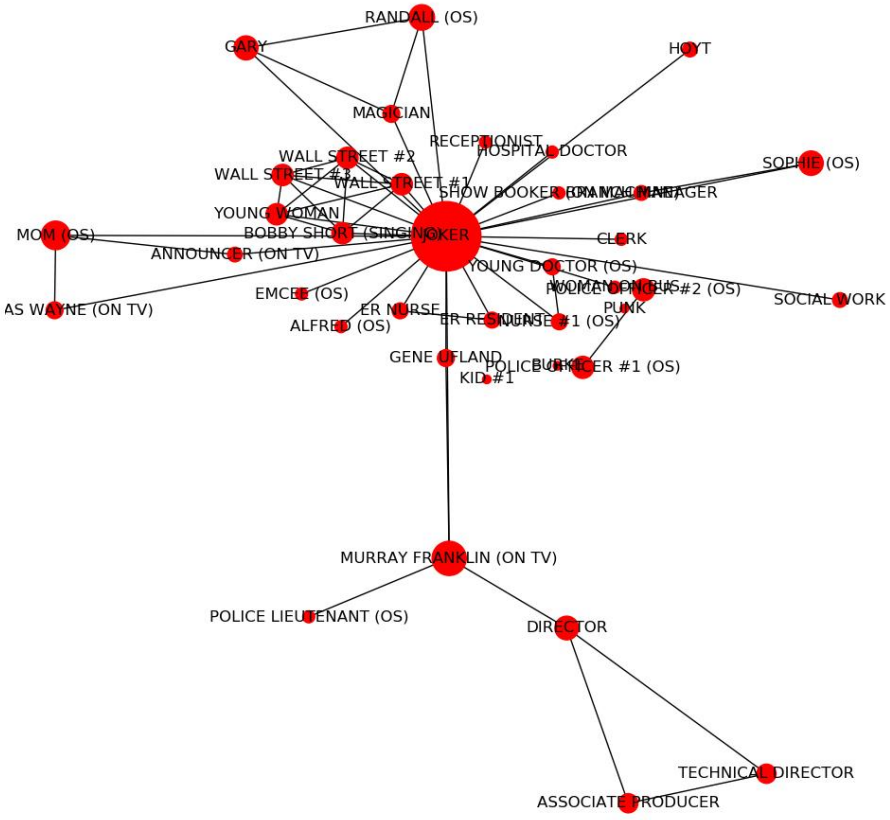

Figura 6 PageRank

3.7 Grupos y comunidades

En este apartado se va a hablar de la distribución de las redes en grupos y comunidades. En todas las redes que podemos generar, siempre hay agrupaciones de nodos que están más

relacionados entre sí que con otros nodos de la red; estos nodos más densamente conectados entre sí que con el resto de la red son los que van a formar las distintas comunidades. Un ejemplo de comunidades que nos resulta familiar a todos podría ser el siguiente: en un instituto en el que los alumnos se relacionan unos con otros, siempre se forman grupos de amigos que aunque no dejan de estar conectados con el resto de alumnos, forman un grupo más estrechamente relacionado; otro ejemplo sería el de una empresa en la que sus empleados se relacionan unos con otros pero existen departamentos en los que los miembros de dicho departamento tienen más relación entre ellos que entre miembros de otros departamentos.

Para poder detectar los distintos grupos y comunidades que se van a formar en una determinada red, podemos emplear distintas métricas y algoritmos.

Cliques

Un clique se corresponde con un conjunto de nodos dentro de una red que están completamente conectados unos con otros, es decir entre cada par de nodos de dicho clique, existe un enlace que lo une. 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(22).

K-clique

El K-clique se diferencia del clique en que los nodos que pertenecen al conjunto no necesitan estar directamente relacionados unos con otros, sino que se permite que los nodos pertenecientes al conjunto estén a una distancia menor o igual que k (23).

K-clan

El k-clan se considera una alternativa al k-clique y dice que los caminos que unen a los nodos del conjunto tienen que formar parte del propio conjunto(24).

K-plex

Un conjunto de *n* nodos va a ser considerado k-plex siempre y cuando se cumpla la condición de que cada nodo del conjunto tiene que estar conectado a *n-k* nodos que formen parte del mismo conjunto (25).

K-core

El k-core de una red va a ser un conjunto máximo de nodos en el cual todos los nodos que forman parte del conjunto tienen un grado de al menos *k*. Para conseguir sacar los k-cores de una red, debemos ir eliminando repetidamente los nodos que tengan un grado menor que *k*(26).

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

Existen diversos algoritmos con los cuales podemos detectar y obtener las comunidades que hay en una red, el empleo de un algoritmo u otro nos puede dar lugar a obtener distintos tipos de comunidades para la misma red, debido a que cada uno de los algoritmos obtiene las comunidades de una forma distinta.

Para este proyecto se han implementado cuatro tipos de algoritmos de detección de comunidades que son los siguientes:

- Girvan-Newman: El algoritmo de Girvan-Newman para detectar comunidades se basa en la técnica de eliminación de enlaces(27). Lo que hace el algoritmo es:
  - Calcular la centralidad de intermediación de todos los enlaces.
  - Elimina aquel que tenga mayor centralidad de intermediación.
  - Recalcular la centralidad de intermediación de todos los enlaces.
  - Repite los pasos dos y tres hasta que no queden más enlaces.
- Louvain: El algoritmo de Louvain para detectar comunidades es una técnica que se basa en optimizar la modularidad de la siguiente forma (6):
  - Se asigna una comunidad a cada nodo de la red.
  - Se calcula el incremento del modularidad que supondría eliminar el nodo de esa comunidad y meterlo en otra comunidad distinta con la que esté enlazado.
  - Después se crea una nueva red de forma que cada comunidad es un nodo de la nueva red.
  - Se repiten los pasos dos y tres hasta que no haya más incremento del modularidad.
- Clauset-Newman-Moore: El algoritmo de Clauset-Newman-Moore es un algorimto muy similar y anterior al algoritmo de Louvain, también basado en el incremento de la modularidad con una diferencia respecto de Louvain, el algoritmo CNM conecta comunidades cuya fusión produce un aumento de la modularidad sin optimizar la modularidad local de todos los nodos primero como sí hace Louvain (7).
- Comunidades K-clique: Este algoritmo emplea el método de percolación de cliques, método con el cual se identifican los k-cliques de la red y se identifican las comunidades, una comunidad se considera cuando hay una unión máxima de todos los k-cliques(8).

3.8 Detección de roles

En este apartado se va a tratar la detección de roles dentro de la red. De acuerdo con el algoritmo de Guimerà et al.(28), cada nodo de la red va a tener un rol asignado, el cual es calculado mediante dos fórmulas explicadas a continuación. Antes de poder calcular los roles dentro de una red es necesario haber realizado la detección de comunidades por alguno de los métodos anteriores. Una vez que se tienen las comunidades, se calculan los roles obteniendo el grado que posee un nodo dentro de la comunidad y su coeficiente de participación. Es de esperar que los nodos se agrupen en los mismos roles si tienes características similares(29).

Para el cálculo se van a necesitar dos fórmulas:

- El grado del nodo dentro de su comunidad, que será calculado de la siguiente forma:

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

Donde tenemos que:

  - Z_i: es el grado del nodo dentro de su comunidad.
  - k_i: es el grado de un determinado nodo i respecto a su comunidad.
  - k_{si}: es la media del grado de todos los nodos a los que pertenece un nodo i.
- Lo segundo que se debe calcular es el coeficiente de participación del nodo de la siguiente forma:

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

Donde debemos tener en cuenta que:

- P_i: es el coeficiente de participación del nodo.
- 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.

Al igual que para la detección de comunidades, en este apartado se ha desarrollado la detección de roles para todas las comunidades detectadas, es decir:

- Girvan-Newman(27).
- Louvain(6).
- Clauset-Newman-Moore(7).
- K-clique(8).

Además, cada rol viene con su respectivo gráfico que se mostrará a continuación 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.

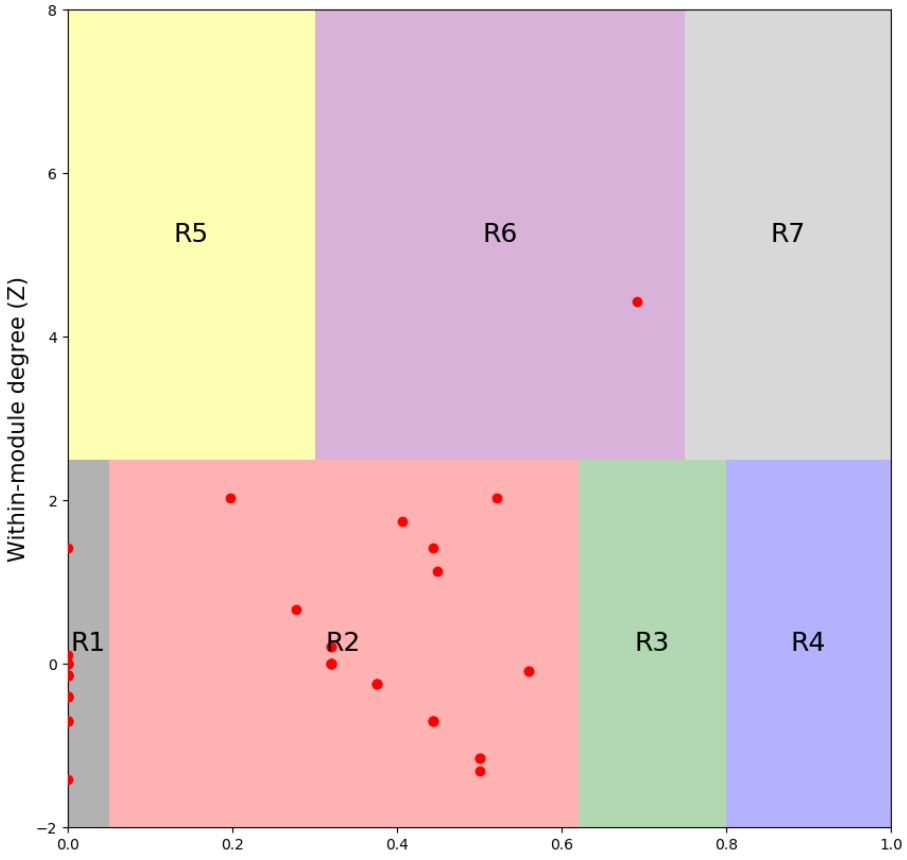

Figura 7 Gráfico de detección de roles

Para asignar cada nodo a su respectivo rol, una vez calculadas la “Z” y la “P” mencionadas anteriormente, se seguirá la siguiente clasificación(28):

- Si la Z 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 P es menor o igual que 0.3. Poseen la característica de que 5/6 de sus enlaces se encuentran dentro de su comunidad.
  - Hubs conectores (R6): son los nodos en los que su P 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 P 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 la Z 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 ultra periféricos (R1): son los nodos en los que su P 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 P 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 P 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 P 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 muy complicados de asignar una comunidad.

3.9 Ethnea y Genni

Ethnea y Genni(9) son unas librerías cuya función es predecir la etnia y el sexo dado un nombre y apellido como input respectivamente. Este software es muy rápido y puede devolver también los valores como json, formato que ha sido el empleado en este proyecto para extraer dichas características.

Además de predecirte el sexo y la etnia principal del personaje, en su aplicación web se puede ver que también da los porcentajes de probabilidad de que el personaje sea de otras etnias distintas a la principal.

3.10 Scraping Web

El *scraping* de una web es una técnica que emplea diversos programas para extraer información de un sitio web. Normalmente lo hace simulando la interacción de un humano con la página (30).

Para este proyecto se ha empleado bastante esta técnica ya que permite sacar grandes cantidades de información de una manera rápida y sencilla. Hay diversas herramientas con las que se puede realizar esta técnica, en este proyecto se ha utilizado la librería de **BeautifulSoup4**.

D. Técnicas y Herramientas

En esta sección se van a explicar las herramientas que han sido usadas a lo largo del desarrollo del proyecto.

4.1 Metodología ágil – Scrum

La metodología SCRUM se basa en el principio ágil de desarrollo iterativo e incremental. Se divide en períodos de trabajo para ir desarrollando el producto final llamados “sprints”, de duración variable dependiendo del proyecto. La metodología SCRUM establece diversas reuniones a lo largo de los “sprints”, una al inicio del “sprint” para establecer el trabajo que se debe realizar y otra al final del “sprint” para evaluar el trabajo que se ha realizado a lo largo del “sprint”, además, se establecen revisiones diarias que son realizadas por el equipo de trabajo en una auto-gestión del proyecto(10).

4.2 Herramienta de control de versiones

- **Herramienta:** GitHub

GitHub es una plataforma para poder albergar proyectos utilizando el sistema de control de versiones de Git(31), todos los repositorios se pueden almacenar de forma pública o privada(32), ha sido elegida debido a la facilidad para visualizar todos los cambios que se han ido realizando en el proyecto y por la integración de ZenHub(33), una herramienta que será explicada posteriormente para la gestión del proyecto.

Otro motivo que ha servido para facilitar la elección de esta plataforma ha sido su utilización previa en asignaturas cursadas a lo largo de la carrera.

4.3 Herramienta de gestión del proyecto

- **Herramienta:** ZenHub

ZenHub es una extensión del navegador empleada para poder añadir funciones adicionales en nuestro repositorio de GitHub de forma que podamos gestionar el proyecto, añadir sprints, tareas especiales sin necesidad de emplear una herramienta externa y de forma rápida y sencilla.(33)

4.4 Herramienta para realización de la documentación

- **Herramienta:** Microsoft Word

Se ha realizado la documentación en Microsoft Word debido al conocimiento que se tiene acerca de este procesador de texto y su facilidad para establecer los distintos formatos de letras, párrafos, encabezados…

Alternativas: Se estudió la alternativa de OpenOffice que fue descartada en seguida ya que para emplear dicha herramienta se prefería realizar la documentación en Microsoft Word. Otra alternativa tanteada para la realización de la documentación fue LaTeX, pero se descartó debido a que LaTeX es una buena elección si tuviéramos que realizar demasiadas fórmulas o expresiones matemáticas para las cuales LaTeX es muy eficaz, pero para la documentación que se iba a realizar se pensó que no supondría demasiada diferencia usar LaTeX o Microsoft Word y con la segunda se tenía mayor experiencia.

4.5 Herramienta para gestionar el repositorio local-remoto

- **Herramienta:** GitKraken

GitKraken es una herramienta que nos permite hacer la clonación de nuestro repositorio de GitHub a una copia local en nuestro ordenador y poder modificar y actualizar este repositorio local sin tener que modificar el repositorio principal de GitHub, además la realización de los *commits* y subida de archivos al repositorio remoto se hace de una forma muy sencilla y rápida (34).

Esta herramienta también ha sido usada a lo largo de la carrera y por eso se ha decidido utilizarla.

Alternativa: Se planteó también la alternativa de GtiHub Desktop (35), una herramienta de GitHub para gestionar el repositorio de la misma forma que lo haría GitKraken, el motivo de no haber escogido esta herramienta es su escasa utilización y conocimiento acerca de ella. Ya que se conoce más el funcionamiento de GitKraken, se empleará esta herramienta.

4.6 Herramienta para la gestión de referencias bibliográficas

- **Herramienta:** Zotero

Zotero es un gestor de referencias libre que posee una extensión para el navegador que permite el almacenamiento de referencias de una forma muy rápida y sencilla, además tiene su propio complemento de Microsoft Word con el cual poder realizar las citas y la bibliografía con gran eficacia y en una amplia gama de formatos.

4.7 *Scraping* Web

- **Herramienta:** BeautifulSoup4 + urllib

BeautifulSoup4 es una librería que permite la obtención de forma sencilla de información de una página web buscando por ejemplo por un determinado tipo de etiqueta html.(36)

Urllib es un módulo de Python para la gestión y trabajo con direcciones web. Tiene cuatro paquetes que son los siguientes(37):

- urllib.request: empleado para abrir y leer direcciones web.
- urllib.error: que contiene las excepciones que son lanzadas por “urllib.request”.
- urllib.parse: empleado para parsear las direcciones web.
- urllib.robotparser: empleado para parsear ficheros “robots.txt”.

En este proyecto se ha empleado el paquete “urllib.request” para abrir la dirección web requerida y la librería BeautifulSoup4 para extraer la información de dicha dirección web.

4.8 Funcionalidad etnia y sexo

- **Herramienta:** Ethena + Genni

Ethnea y Genni(9) es una librería empleada para obtener la etnia y el sexo de una persona simplemente con la introducción de su nombre y apellido de una forma sencilla y rápida. Se intentó implementar la librería en proyecto pero debido a que no se encontró la librería pública se optó por hacer *scraping* y obtener así el sexo y la etnia del personaje, esto se ha hecho debido a que “Ethnea + Genni” permite obtener la información en formato json simplemente añadiendo “&format=json” a la página de tal forma que se obtienen los datos de una forma mucho más sencilla (9) (38).

4.9 Lenguaje de programación

- **Herramienta:** Python

Se ha empleado Python como lenguaje de programación debido a su conocimiento a lo largo de la carrera y a que la migración de todo el proyecto a otro tipo de lenguaje podría ser muy costosa. A su vez, la librería NetworkX de Python, es una de las más utilizadas en todo el mundo para el análisis de redes.

4.10 Generación de la red

- **Herramienta:** NetworkX

NetworkX es una librería de Python que permite la creación, manipulación y análisis de las distintas características que poseen las redes complejas(39). NetworkX posee de forma gratuita una gran cantidad de métricas que se pueden extraer de una red, aunque para algunas características en concreto ha sido necesaria la instalación de una librería adicional complementaria a NetworkX: *Python-louvain* (40) (41).

El motivo principal para la elección de NetworkX es el empleo de esta librería a lo largo de la carrera en algunas asignaturas de forma que ya se conocía de una forma consistente su uso.

4.11 Visualización de la red

- **Herramienta:** network styling with d3

Para la visualización de la red se ha empleado la aplicación web mencionada anteriormente ya que permite una visualización de la red de manera interactiva, con distintas opciones para modificar la red a gusto del usuario. Para la introducción de la red, la aplicación web te pide que introduzcas una dirección a un fichero o que subas un fichero directamente, para facilitar el trabajo al usuario, la aplicación introducirá directamente la red de forma automática.(42)

Netwulf(43) emplea esta misma aplicación web con el fin de visualizar las redes generadas en NetworkX.(44)

Alternativa: La alternativa a esta aplicación web es sacar el grafo como se extrae en el informe, es decir, mediante NetworkX, la problemática que surgía con esto es que NetworkX no nos permite visualizar el grafo de forma interactiva, sino que visualizaríamos una imagen del grafo. No se podría modificar la red a gusto del usuario.

4.12 Analizador léxico

- **Herramienta:** Ply

Ply es una librería que implementa las herramientas de análisis léxico lex y yacc para Python. Permite crear un parser o un lexer siempre y cuando se conozcan las herramientas mencionadas anteriormente de lo contrario podría resultar más costoso.

Uno de los motivos de su utilización es el empleo de herramientas similares a lo largo de algunas asignaturas del grado.

4.13 Interfaz gráfica

- **Herramienta:** Flask

Flask es un framework escrito en Python para poder desarrollar aplicaciones web de una forma sencilla y rápida empleando un número reducido de líneas de código. Flask está basado en

la especificación WSGI (Web Service Gateway Interface) de Werkzeug, que sirve para que los servidores web puedan enviar solicitudes a aplicaciones web desarrollados en el lenguaje de programación de Python, y el motor de plantillas Jinja2 (45) (46).

Además de lo anteriormente mencionado, Flask posee una extensión llamada Flask Babel que permite la internacionalización de una forma sencilla de la web (47).

4.14 Herramienta para la interfaz

- **Herramienta:** Bootstrap4

Para la realización de la interfaz, en vez de realizar una interfaz partiendo de 0 lo cual probablemente iba a llevar demasiados costes de tiempo y probablemente no se iban a obtener los resultados esperados, se optó por implementar la herramienta de Bootstrap.

Bootstrap es una biblioteca multiplataforma de código abierto que permite a los desarrolladores diseñar sus sitios y aplicaciones web(48). Una de las ventajas por la cual se eligió esta biblioteca es que posee varias plantillas con las cuales diseñar tu web de una forma rápida y sencilla, además de tener una documentación bastante completa al respecto(49).

4.15 Programación en el lado del cliente

- **Herramienta:** JavaScript + jQuery

JavaScript es una lenguaje de programación interpretable por cualquier navegador actual para poder modificar las páginas web por la facilidad de interacción con el DOM a través del cual se podrán suprimir, añadir o modificar los elementos de la web(50).

JQuery es una biblioteca de JavaScript que simplifica la forma de interaccionar con los documentos HTML, manipular el árbol DOM, controlar eventos, y añadir interacciones con la técnica AJAX, que sirve para crear aplicaciones interactivas que serán ejecutadas desde el cliente, pero manteniendo su comunicación con el servidor en segundo plano, de tal manera que es posible que se realicen cambios sobre la página sin necesidad de que esta sea recargada(51), a páginas web(52).

4.16 Realización de la wiki

- **Herramienta:** Wikidot

Wikidot es un servicio de alojamiento para wikis con el cual permitir la realización de una wiki propia sin necesidad de descargar ningún tipo de archivo ni instalar ningún componente. Wikidot tiene versión gratuita en la cual permite el uso de cinco wikis distintas, todas ellas con un número máximo de espacio el cual puede ser ampliable cogiendo la versión de pago de wikidot (53).

Alternativa: Se experimentó con MediaWiki, ya que es la wiki empleada por Wikipedia, con una amplia documentación acerca de las diversas alternativas que ofrece (54). Finalmente se decidió no utilizarla debido a un problema que surgió al desplegar MediaWiki en Heroku, ya que con la versión gratuita de Heroku, tras 30 minutos de inactividad la aplicación se iba a “dormir”, y al ocurrir esto, todas las imágenes que habían sido subidas a la wiki se perdían. De todos modos, sigue desplegada y se puede visitar accediendo a https://wikinetextractor.herokuapp.com.

4.17 Herramienta para albergar la aplicación

- **Herramienta:** Heroku

Heroku es una plataforma como servicio para poder desplegar aplicaciones web en la nube pudiendo tener tu propio espacio para albergarla de forma gratuita y en varios lenguajes de

programación. Además, tiene diversas opciones de pago que permiten ampliar el almacenamiento de la web y evitar ciertas restricciones que posee el despliegue gratuito como que las aplicaciones hacen un “sleep” después de 30 minutos de inactividad. Posee una gran ventaja ya que el despliegue es muy rápido y sencillo dejando al desarrollador preocuparse del desarrollo de la web y no del despliegue, además tiene una gran escalabilidad para las aplicaciones desplegadas en esta plataforma(55).

Alternativa: Se exploraron otras alternativas como OpenShift descartadas después de comprobar que la versión gratuita incluía características parecidas a Heroku pero su despliegue parecía más complicado (56).

4.18 Herramienta para gestionar las traducciones

- **Herramienta:** Poedit

Para gestionar la internacionalización de la web, se empleó la aplicación de Poedit (57), la cual permitía abrir el catálogo de traducciones con el fin de asignar una traducción en inglés a las frases y palabras que estaban en castellano a lo largo de la aplicación.

4.19 Herramienta para generación de diagramas

- **Heramienta:** Draw.io

Draw.io es una herramienta que permite generar de una manera cómoda y sencilla una gran gama de diagramas UML, además, posee una serie de plantillas, que pueden ayudar con la realización de los diagramas.

Es una herramienta gratuita que no necesita ninguna descarga adicional por lo que simplemente accediendo al enlace anterior se puede empezar ya con la creación de los diagramas UML.

4.20 Herramienta para la creación de prototipos

- **Herramienta:** Adobe XD

Adobe XD(58) es una aplicación de escritorio la cual permite la creación de prototipos para aplicaciones web y para aplicaciones móviles. El tamaño que queremos asignar a la pantalla es muy personalizable y posee una gran cantidad de complementos para hacer que los prototipos sean mucho más completos.

E. Aspectos relevantes de desarrollo del proyecto

En este apartado se va a hablar de las diversas tomas de decisiones que se han llevado a lo largo del desarrollo de la aplicación y puntos críticos en el desarrollo, así como de problemas que han surgido en la realización del proyecto.

5.1 Inicio del proyecto

La idea de la realización del proyecto nació del interés por las redes complejas que surgió cuando se cursó la asignatura de "Nuevas Tecnologías", cuyo contenido tenía que ver con el mundo de las redes complejas, sus características y el análisis de sus propiedades.

Cuando apareció la opción de continuar el proyecto de Ububooknet para la creación de redes a partir de novelas realizado por Luis Miguel Cabrejas Arce, cumpliendo el estilo de TFG deseado, al que le faltaban aún de implementar varias funcionalidades, nos pusimos en contacto con el fin de formalizar el proyecto.

Una vez formalizado y con la aprobación recibida para la realización del proyecto, comenzó el desarrollo de la aplicación.

5.2 Metodologías

Se ha seguido una metodología SCRUM basada en el desarrollo incremental del proyecto, aunque debido a la realización del proyecto en un ámbito individual (sólo había una persona desarrollando el proyecto) hay ciertas diferencias con la metodología SCRUM habitual, aun así, se ha intentado seguir en la medida de lo posible las características de la metodología:

- Se ha diseñado la aplicación de una forma incremental.
- Se han mantenido reuniones al principio de cada *sprint* con el fin de establecer las funcionalidades que se debían añadir.
- Se han mantenido reuniones al final de cada *sprint* para hablar de cómo ha ido el *sprint* y de todo lo que se ha implementado.
- Se han ido realizando revisiones diarias a modo de auto-gestión para ir comprobando el estado del *sprint* y de la aplicación en cada momento.
- Con la herramienta de ZenHub se han ido moviendo las tareas una vez que se completaban para ir eliminando trabajo por hacer.
- Una vez se tenían las funcionalidades que se debían añadir en un *sprint* concreto, se estimaba la duración que iban a tener.
- Se ha respetado un tiempo para cada *sprint* de aproximadamente 2 semanas.

5.3 Formación

El lenguaje de programación en el que se desarrolla el proyecto es Python, lenguaje del cual se ha trabajado en multitud de ocasiones a lo largo de la carrera. No obstante, algunas herramientas que se han empleado para la realización del proyecto eran desconocidas o no se había trabajado nunca con ellas, es por eso que un peso importante del proyecto consistió en documentarse sobre ello, leyendo manuales y documentación al respecto o realizando tutoriales y guías con las cuales aprender a manejar dichas herramientas.

Bootstrap

Para la realización de la interfaz gráfica, se pensó que lo mejor sería implementar una posible plantilla en vez de intentar desarrollar toda la web y sus hojas de estilo de forma manual, para lo cual se recurrió a Bootstrap debido a que ofrece una multitud de plantillas las cuales se pueden descargar e implementar de una forma sencilla.

Como no se había trabajado nunca con Bootstrap, antes de empezar a desarrollar la interfaz, se hizo la lectura de una guía que ofrece Bootstrap con todas las gamas de opciones que se pueden realizar a través de esta herramienta. El acceso a la guía se puede realizar mediante el siguiente enlace:

- https://getbootstrap.com/docs/4.0/getting-started/introduction/(49).

Scraping Web

En cuanto a la extracción de datos de los guiones tampoco se conocía bien la estructura por tanto hubo que acceder a la página que los ofrecía y explorar como se distribuían las escenas, personajes… Aquí ha habido varios problemas con algunos guiones, ya que su obtención se realiza mediante *scraping*, pero no todos los guiones siguen el mismo formato, por tanto, algunos guiones expuestos en la web https://www.imsdb.com/ no son capaces de obtenerse, se intentó solventar en la medida de lo posible pero dado que los guiones son subidos por diferentes usuarios sin respetar una estructura estándar, se descartó esta opción porque era muy demandante en términos de tiempo y no aportaba un valor tan relevante como para justificar semejante inversión de tiempo.

Interacción aplicación-usuario

La aplicación tiene botones de navegación que permiten el avance o retroceso por la aplicación, cuando se inició el proyecto había ciertas pantallas que permitían el acceso al usuario a pesar de no haber realizado pasos previos necesarios para poder avanzar, un ejemplo de ello era la fase de creación de diccionarios, la cual permitía el avance del usuario a su modificación, sin necesidad de introducir ningún diccionario.

Es por esto que, en esta versión de la aplicación, se ha mejorado la interacción de la aplicación con el usuario de forma que el usuario pueda saber siempre, a través de alertas mostradas en pantalla, el por qué no puede acceder a alguna de las pantallas de forma que el usuario pueda realizar los requisitos necesarios para poder avanzar a la pantalla siguiente.

Flask

En este proyecto se trabaja con Flask, un *framework* escrito en Python que permite la creación de aplicaciones web de una manera muy rápida y sencilla, posee multitud de extensiones que pueden implementarse para hacer crecer cualquier proyecto como Flask Babel, extensión utilizada en el proyecto para su internacionalización. Además, Flask aporta al proyecto una estructura sólida siguiendo el patrón de MVC que se aplica en el proyecto por el cual la estructura lógica de la aplicación (Modelo) está conectado a la interfaz (Vista) a través de un intermediario (Controlador). Se puede modificar la estructura de la aplicación sin tener que alterar la interfaz y viceversa. También, Flask permite probar lo que se está desarrollando en un entorno propio sin tener que recurrir a un servidor web para realizar las pruebas.

Para su utilización, se siguieron los tutoriales recomendados en el proyecto anterior (Ububooknet):

- Flask Tutorial Web Development with Python(59).
- The Flask Mega Tutorial(60).

5.4 Desarrollo de los algoritmos

Antes de comenzar el apartado de desarrollo de algoritmos vamos a establecer un punto de partida desde el que se empezaban a desarrollar los algoritmos. En este proyecto tenemos que diferenciar entre la obtención de personajes, apariciones y enlaces para una novela (ePub, ya implementada en el proyecto anterior) y para una película (introducción del guion obtenido de la página http://www.imsdb.com). Para la obtención de personajes por ePub ya se había descrito que los personajes comenzarían con mayúscula, para la obtención de las apariciones simplemente se cuentan todas las veces que aparece ese personaje en el texto y para la obtención de enlaces se consideró una interacción entre dos personajes si aparecían dos nombres en un intervalo de palabras hacia adelante y hacia atrás establecido por el usuario.

Al tener otro sistema de obtención de datos, es preciso diseñar nuevos algoritmos con los cuales obtener estos valores. Estos algoritmos son los que se van a explicar a continuación:

a. Obtención del diccionario de personajes:

El principal va a ser la obtención de los personajes, para la cual necesitábamos realizar un barrido por la web cogiendo los nombres que se correspondieran con personajes del guion, aquí apareció el primer problema ya que el algoritmo leía la etiqueta correspondiente a los personajes, dicha etiqueta se correspondía también con textos que no tenían nada que ver con nombres de personajes y párrafos enteros cuya etiqueta era la misma. Por lo tanto, se propuso hacer un "filtro" para eliminar ciertos caracteres que no se corresponderían con unos personajes:

- El primer filtro fue establecer un número máximo de caracteres, cabe pensar que el nombre de un personaje no va a ser mayor de 25 caracteres, para ello se leyeron un número considerable de guiones comprobando si podría darse un personaje de tal magnitud y se llegó a la conclusión que no. Además, muchos nombres de personajes en la página están acortados como "Darth Vader" quien solo figura como "Vader".
- Lo segundo que se filtró fueron los distintos símbolos que no se corresponden con nombres y que en alguna ocasión aparecían como pueden ser "*, @, $, %, ...". Esto lograba acortar en gran medida la lista de falsos positivos, aunque no se reducía al máximo ya que hay ciertos tecnicismos de algunas escenas que tienen el mismo formato que los personajes, para ello hay una función de borrar personajes.
- Se pensó también en la posibilidad de eliminar los números de los nombres, pero fue una opción que se descartó rápido debido a que, en muchos guiones, personajes secundarios como niños que aparecen en ocasiones contadas a lo largo de la película, son llamados "Kid #1" o "Kid #2" con lo cual si quitáramos los números se eliminaría un gran número de falsos positivos, pero a su vez eliminaríamos ciertos personajes que harían que la red no fuera completa.

b. Obtención de las apariciones de los personajes:

El segundo algoritmo se corresponde con la obtención de las apariciones de los personajes, para obtener las apariciones, se dedujo que, como las películas y sus guiones se agrupan en escenas, un personaje que apareciera cinco veces en la misma escena solo debía ser considerado como una aparición. Por tanto, el algoritmo lo que hacía era aumentar en uno el contador de apariciones por cada escena, las escenas están distribuidas y separadas por "INT" o "EXT" en la mayoría de los guiones, excepto en algunos concretos que no hay distinción entre escenas como mencionamos anteriormente. Así que, el algoritmo va recorriendo la página y

entrando en cada una de las escenas, cuando lee el nombre que está buscando aumenta en uno el contador y ya no lo aumenta hasta que encuentra dicho personaje en una escena distinta.

c. Creación de enlaces

En cuanto a la creación de enlaces, ya que estamos realizando redes de interacciones entre personajes, lo primero que se hizo fue definir el concepto de lo que se iba a considerar interacción. En la parte de las novelas ya se había definido interacción teniendo en cuenta un intervalo de palabras que debía existir entre cada personaje, pero en el caso de los guiones era distinto, debido a que ya no teníamos un número de palabras entre personajes, sino que teníamos nombres de personajes con su respectivo texto.

Para ello se llegó a la conclusión que una interacción entre dos personajes existía en el momento en que aparecían juntos en la misma escena, de tal manera que, si dos personajes aparecían cuatro y dos veces en una misma escena, contaría como una interacción, es decir, sumar uno al peso del enlace entre ambos personajes. De esta forma se le asignaría el peso al enlace que hay entre dos personajes.

Para conseguir saber estos datos, existe un diccionario de apariciones en el cual la clave es el id del personaje y el valor es una lista con todas las escenas en las cuales aparece dicho personaje. En caso de que un personaje tenga más de un nombre, se juntará a la id las escenas de cada uno de los nombres.

d. Ethnea y Genni

El último algoritmo implementado en la parte final del proyecto fue el de Ethnea y Genni(9) ya que parecía óptimo poder implementar una librería que fuese capaz de predecir que etnia y sexo serían los personajes que aparecen.

Para ello se buscó la librería que se necesitaba para implementarlo y, ante la situación de no encontrarse dicha librería y la falta de tiempo que ya empezaba a surgir, se optó por hacer *scraping* a la web que obtenía las predicciones, en un principio parecía más complicado ya que la web tenía varias etiquetas que podían causar ciertos problemas a la hora de sacar los datos pero posteriormente se encontró una opción para poder visualizar simplemente el json con las predicciones de tal forma que era mucho más sencillo obtener dichas posiciones del json.

Otro dilema que surgió era que, para predecirlo, era necesario introducir nombre y apellidos, pero había personajes que solo estaban nombrados en el guion por un solo nombre. Para solucionarlo se diseñó el algoritmo de tal forma que, si el nombre que aparecía no contenía apellidos sino nombre únicamente, se metiera el nombre como nombre y como apellidos, no es la solución óptima, pero es la única solución viable teniendo los datos de los que se disponía.

El último problema que surgió fue con las herramientas de *scraping*, en concreto con urllib, debido a que había caracteres como “#” en el ejemplo de “KID #1” que no podía leer la librería. Para solucionarlo se creó un algoritmo que cogía los caracteres que no se podían leer y los sustituía por “” de tal forma que el nombre del personaje seguía siendo el mismo, pero en este caso a la hora de introducir los nombres en la librería figuraría nombre KID, apellido 1, sin la almohadilla.

5.5 Arquitectura MVP

Para esta aplicación se ha continuado con la arquitectura MVP o Modelo-Vista-Presentador, la cual permite separar la lógica de la aplicación de la interfaz que tiene dicha aplicación, de tal forma que podemos modificar ciertos aspectos de la lógica de la aplicación sin tener que cambiar nada de la interfaz (61).

La arquitectura funciona de la siguiente manera: la vista sería la interfaz, que son las plantillas HTML, las cuales informarán al presentador de los cambios que está realizando el usuario; el presentador recoge esa información y la pasa al modelo para que realice las acciones que sean necesarias; a continuación, el modelo informa al presentador de que las acciones han sido realizadas, y éste le muestra al usuario los cambios (vista).

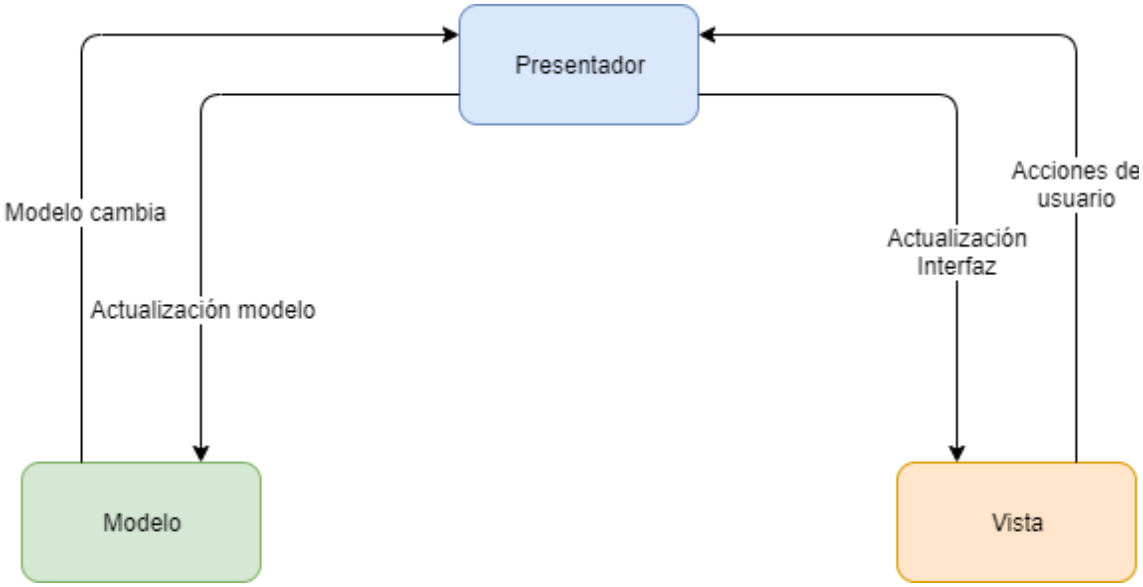

Figura 8 Arquitectura MVP

5.6 Detección de comunidades y roles

Otro cambio relevante es la adición de nuevos roles y comunidades, ya que, si el usuario quería consultar por ejemplo los roles, de dos algoritmos de detección de comunidades que existían, solo obtendría los roles de uno de ellos, un cambio importante que se implementó es la detección de comunidades y roles con cuatro algoritmos distintos.

Además, en la detección de roles se añade un gráfico con el cual se puede observar de una forma más gráfica y quizá más entendible cómo es la distribución de los nodos por los distintos roles presentes en las redes complejas.

5.7 Testeo de la aplicación

Para testear muchas de las funciones que hay en la aplicación como añadir personajes, borrarlos, añadir etnia, añadir sexo… se ha diseñado una clase de Python en la cual se realizan todas estas funciones para comprobar que todas funcionan de manera correcta.

Un gran problema para testear funciones de obtención de personajes de los guiones es que no sale el valor de cuantas apariciones debe haber por cada personaje, cosa que complica testear si ese “78” en el número de apariciones de cierto personaje es real o no. Para esto se ha tenido que contar a mano en algunas películas las apariciones de algunos personajes a ver si lo hacía bien o mal esto supone un problema respecto a que no se ha podido hacer un testeo automático de algunas características.

Betatesters

Encontrar errores en una aplicación diseñada por uno mismo es más complicado de lo habitual, debido al conocimiento que hay de la aplicación y de todas las opciones que existen; por ello, se ha recurrido a un grupo de voluntarios para que prueben la aplicación y vean si encuentran posibles errores, con el fin de poder solucionarlos antes del lanzamiento oficial de la aplicación.

Cuando se consideró que la aplicación estaba “lista” para ser lanzada, se compartió con los voluntarios y se les pidió que escribieran sus impresiones y los posibles errores que identificaban. Todo esto fue recogido en el siguiente documento: Comentarios *Betatesters*.

Gracias al *feedback* de estos voluntarios, se resolvieron algunos problemas que tenía la aplicación. Algunos de los principales problemas solucionados fueron:

- Uso en móvil: la aplicación tenía una estructura muy definida para una pantalla más grande y la visualización de la red era demasiado ineficiente, esto se solucionó arreglando las plantillas html y reestructurando algunos elementos, de tal forma que la aplicación se volvió más amigable al uso de dispositivos móviles.
- Error 500: ciertos guiones que no cumplen la estructura especificada en el manual de usuario de la aplicación causaban que, a la hora de introducirlos a la aplicación, mandaran un Error 500 (*Internal Server Error*) y se saliera de aplicación. Esto se solucionó añadiendo filtros para que, si el guion no cumple la estructura solicitada, alerte al usuario de ello sin salir de la aplicación.
- Duplicación de posiciones: se detectó también un error que hacía que cuando se calculaban las posiciones de nuevo, se duplicaran las apariciones que tenía un personaje, también que cuando se juntaban dos personajes y se recalculaban sus posiciones, difería el resultado del número esperado. Esto se solucionó cambiando parte del algoritmo de obtención de posiciones, siempre que se recalcularan las posiciones, se debía de reiniciar el número de apariciones.
- Cambios menores: otros cambios que se detectaron en la aplicación se corresponden con ligeras variaciones de la misma, correspondientes sobre todo al tema de malas traducciones y traducciones no realizadas.

5.8 Problemas derivados del código

Uno de los principales problemas que se ha considerado relevante y por lo tanto se va a explicar es la dificultad de haber trabajado sobre código ya realizado. Supuso un grave problema y un gran coste de tiempo al inicio del proyecto entender todas las cosas que se realizaban en la aplicación.

Se cogió una aplicación en la cual no funcionaban algunos botones debido al problema de sesiones, que se describió en los objetivos generales de esta memoria, por el cual no podíamos avanzar por la aplicación sin que nos diera un error de sesión, cerrando sesión y, por tanto, volviendo a la página de inicio.

Por este motivo, se dedicó el primer sprint exclusivamente a leer y entender toda la documentación y el código de la aplicación para conseguir entenderlo, con todo ello, en los sprints posteriores aún hubo que ojear de vez en cuando la documentación por nuevos problemas que surgían y que había que solucionar.

5.9 Servidor de alojamiento de la aplicación

El servidor elegido para alojar la aplicación es Heroku, una plataforma que permite el *hosting* de aplicaciones web en diferentes lenguajes de programación. Es una herramienta muy sencilla de usar, con un despliegue muy rápido e intuitivo.

El problema de este servidor es que, al ser una versión gratuita, tiene ciertas limitaciones. Cada media hora, el servidor hace un *sleep*, es decir, la aplicación se queda parada a la espera de que alguien vuelva a acceder a ella. Esto supone un coste de tiempo al intentar acceder después de un período de inactividad.

Quizá el problema más reseñable es respecto a los *timeouts*, Heroku lanza un *timeout* en la aplicación si se llega a un límite de 30 segundos de ejecución. Esto es un problema que podríamos solucionar alojando la aplicación en un servidor propio. Debido a esto, cuando se quiere obtener la etnia y el sexo de los personajes, Heroku lanza un *timeout* y se bloquea en "cargando". Para poder continuar con el uso de la aplicación, debemos de refrescar la página pulsando "F5".

5.10 Wiki

Otro aspecto relevante de la aplicación es la realización de una wiki. A pesar de que la creación de una wiki puede suponer un costo de tiempo elevado, ya que hay que escribir toda la información que se muestra al usuario en la aplicación, y explicarla paso a paso; supone una amplia mejora para la usabilidad de esta, es una práctica muy habitual hoy en día en el desarrollo de aplicaciones, ya que supone una herramienta que complementa la aplicación, permitiendo al usuario facilitar su entendimiento.

A nivel personal también supone un aprendizaje a la hora de su desarrollo, investigación de herramientas que ofrecen este tipo de servicios y despliegue de la misma como es el caso de MediaWiki, a pesar de que luego no se acabara implementando. Además, la elaboración de una wiki supone una funcionalidad adicional en el proyecto que lo enriquece, haciéndolo más completo para el usuario y mejorando la usabilidad.

Para su realización se pensó en un principio en MediaWiki, incluso se llegó a desplegar MediaWiki en Heroku como se puede comprobar si se accede al siguiente enlace:

- https://wikinetextractor.herokuapp.com.

El problema surgió con que MediaWiki necesita una base de datos en la cual guardar temas como imágenes de la wiki. No se llega a saber concretamente si es por usar la base de datos gratuita que da Heroku o si es porque el despliegue gratuito pone la aplicación en "sleep" después de treinta minutos de inactividad, pero al cabo de cierto tiempo las imágenes desaparecen de la wiki, como si se hubieran borrado.

La causa se cree es debida a que al usar la parte gratuita de Heroku, tanto para el despliegue de la wiki como para la base de datos, hay ciertas restricciones que hacen que se eliminen las imágenes de la wiki. De todos modos, si se consiguiera un servidor externo a Heroku se podría desplegar la wiki de forma sencilla.

5.11 ICEUTE (*International Conference on EUropean Transitional Eductaion*)

Vivimos en el mundo de la interconexión y el Big Data, por ello, decidimos diseñar NetExtractor con el fin de hacer accesible a todo el mundo el conocimiento de teoría de redes. Precisamente por este motivo, creemos que NetExtractor tiene un enorme potencial docente, pudiendo ser especialmente útil además de para acercar la teoría de redes a los estudiantes, para mostrar que las matemáticas no son solo números y cálculos, sino que son fundamentalmente pensamiento abstracto y formalización. Esto lo conseguimos mediante el uso de ejemplos que al estudiante pueden resultarle muy familiares como pueden ser películas o novelas, de forma que le resulte más fácil situar los personajes (nodos) de la red y ver las relaciones que entre ellos aparecen.

Por todo lo anterior, este proyecto va a ser presentado en el congreso titulado *The 11th International Conference on EUropean Transnational Educational* -ICEUTE- 2020 (62), congreso en el que se debatirán y presentarán los últimos avances y trabajos sobre la educación

superior en países europeos, organizado por la Universidad de Burgos, GICAP, ITCL y la Universidad de Salamanca que tendrá lugar este año 2020 en Burgos.

F. Trabajos relacionados

En los últimos años ha habido diversidad de trabajos acerca de la ciencia de redes y como se extraían estas mismas de multitud de entornos, desde casos reales de extracción de redes en institutos hasta casos más ficticios como los que estamos trabajando actualmente con las novelas y películas.

Algunos ejemplos de trabajos relacionados con el proyecto de NetExtractor van a ser los mencionados a continuación.

6.1 Ububooknet

Debido a que este proyecto surgió como una continuación del mencionado en este apartado, Ububooknet(63), se considera que este trabajo debe aparecer el primero de los trabajos mencionados.

Este proyecto consiste en una aplicación web en la cual se obtienen, visualizan y analizan redes creadas a partir de novelas que son introducidas en forma de ePubs.

La obtención de personajes se realiza mediante la lectura del ePub con un analizador léxico de tal manera que las palabras que aparezcan en mayúscula su primera letra se considerará un personaje. Esta medida produce un gran número de falsos positivos, es por esto que además la aplicación incorpora una forma de obtener diccionarios a través de una wiki de fandom o la posibilidad de importar los diccionarios a través de un fichero csv que contiene los personajes.

En este proyecto las interacciones de personajes ocurren cuando aparecen dos personajes dentro de un rango de palabras, pero este rango no es estático, sino que es el usuario el que se encarga de introducir el número de palabras máximo al que se deben encontrar dos personajes para ser considerado interacción. Además de establecer este valor, al usuario se le permite introducir las apariciones que quiere que tenga mínimo un personaje, si se quieren tener los capítulos en cuenta o no para obtener los personajes y diversas opciones para modificar el diccionario de personajes (añadir personaje, borrar personaje, modificar ids…).

Finalmente, el usuario podrá visualizar la red de personajes para lo cual se implementó Netwulf(43) una herramienta de visualizado de redes que permite además modificar ciertos parámetros para mejorar la visualización. Y por último podrá visualizar las características seleccionadas en un informe final.

6.2 Network of Thrones

En *Network of Thrones* (64), Andrew Beveridge y Jie Shan analizaron el tercer libro de juego de tronos “Tormenta de espadas” para transformar la novela y serie de éxito en una red de interacciones entre los distintos personajes de la novela.

Con este trabajo se acabó logrando una red que posee 107 nodos, que son los que representan a los personajes de la novela y un total de 353 enlaces que unen los nodos. Estos enlaces tenían asignado un peso que se incrementa cada vez que se produce una interacción entre dos personajes.

Para ello se define el concepto de interacción como los personajes que se encuentran a no más de 15 palabras de diferencia entre ambos, elegido por los autores por considerarse un rango adecuado para una novela. De esta forma se establecieron las relaciones más fuertes cuando el enlace tenía un mayor peso.

Después, se obtuvo la red de personajes en la cual se analizaron la importancia que tienen estos personajes en la red, implementando las medidas de centralidad más importantes y comunes que existen. También se hizo un análisis de las comunidades que se pueden obtener mediante el algoritmo de Louvain (6) que también se emplea en nuestro proyecto.

6.3 Who Is the Most Important Character in Frozen?

Petter Holme, Mason A. Porter y Hiroki Sayama autores de *Who Is the Most Importatn Character in Frozen?*(65) nos van a mostrar, a través de la película de Frozen, las distintas medidas de centralidad que se pueden emplear para la extracción de características de una red de forma que se pueda demostrar lo siguiente:

- Las medidas de centralidad pueden ser utilizadas no solo para la película de Frozen, sino de forma general en cualquier red compleja que exista, por muy distintas que sean la clase de los nodos.
- Que dependiendo de la medida de centralidad que estemos usando, el resultado puede ser totalmente distinto y los nodos importantes pueden cambiar. De manera que lo importante es la pregunta a la que se responde para poder realizar el modelado de la red.

6.4 Evaluating named entity recognition tools for extracting social networks from novels

Niels Dekker, Tobias Kuhn y Marieke van Erp son los autores de este trabajo(66) que consiste en la extracción de características en redes complejas extraídas de novelas. Para la realización de este trabajo, se seleccionaron cuarenta novelas, de las cuales veinte de ellas son novelas clásicas y otras veinte son consideradas novelas modernas.

Para la detección de los personajes se borraron manualmente algunas secciones de las novelas como los agradecimientos, notas de la editorial, información del autor, apéndices y glosario, números de página, títulos de los capítulos, índices y opinions de otros escritores, de forma que se pueda centrar la detección de personajes, únicamente en la trama. A su vez, se definió el término de interacción, considerando la misma como la aparición de dos personajes en la misma frase.

De esta forma se pudieron crear las redes de personajes extrayendo las siguientes cararcterísticas de ello:

- Grado medio de la red.
- Grado medio de la red teniendo en cuenta el peso de los enlaces.
- Longitud de camino media.
- Diámetro.
- Densidad.
- Modularidad.
- Componentes conectados.
- Coeficiente de *clustering* medio.

Una vez extraídas las características, se concluyó el estudio comparando las distintas medidas en las novelas modernas y en las novelas clásicas y estudiando las diferencias. Se concluyó que la diferencia entre ambos grupos era mínima, existiendo la densidad, como la única característica más diferenciadora entre ambas, observando que las novelas clásicas eran más densas por lo general que las modernas.

G. Conclusiones y líneas de trabajo futuras

En este apartado se van a tratar las conclusiones obtenidas de la realización de este proyecto y las líneas de trabajo futuras pensadas para continuar con el desarrollo de la web.

7.1 Conclusiones

Finalmente, se han cumplido todos los objetivos que se pedían cuando se inició el proyecto, se ha conseguido añadir funcionalidad al informe, de tal forma que se obtienen un mayor número de características, se ha añadido una funcionalidad principal extra, ya que ahora podemos obtener redes de guiones de películas también, se han añadido predictores de etnia y sexo para los personajes y se permite al usuario modificar esos campos en caso de que no sean correctos, se han añadido estos datos como atributos del nodo y se ha cambiado toda la interfaz de manera que sea más visual para el usuario y más sencilla de trabajar.

En cuanto a los objetivos personales, se pueden considerar cumplidos, he obtenido más conocimientos acerca de herramientas que eran desconocidas, y he aprendido bastante sobre desarrollo web. A su vez, he ganado experiencia con la metodología SCRUM, la cual había sido implementada en menor medida en algunas asignaturas.

Se ha conseguido realizar una aplicación con una red personalizable en la que cada usuario puede hacer y visualizar la red como prefiera y extraer las características deseadas, además de permitir descargar la red en distintos formatos de manera que pueda ser visualizada en otras herramientas de redes complejas tales como Gephi, R u otras.

Esta aplicación consigue que cualquier usuario pueda utilizarla sin ser un experto en programación ni en teoría de redes. A su vez, la wiki sirve de soporte y referencia para hacer consultas sobre todas aquellas dudas que puedan surgir.

7.2 Líneas de trabajo futuras

En este apartado se van a tratar los trabajos futuros que se pueden realizar sobre la aplicación NetExtractor: las funcionalidades adicionales que se le pueden añadir y algunas soluciones a errores que se pueden tener en cuenta.

Alojamiento de aplicación y wiki

En primer lugar, sería óptimo de cara al futuro buscar un servidor donde alojar la aplicación sin tener que estar pendiente de que la aplicación haga un “sleep” después de treinta minutos de inactividad.

De la misma forma, lo ideal sería dejar de lado wikidot e implementar MediaWiki en un servidor propio u otro servicio que no haga ningún “sleep”.

Gráfico de detección de roles

Cuando se visualiza el gráfico de detección de roles, se hace de forma que los nodos se representan como puntos que se distribuyen a lo largo de los roles existentes, pero no se diferencia ni se señala qué punto corresponde a cada nodo. Una mejora interesante sería poder ver con qué nodo se corresponde cada punto.

Detectar personajes

Implementar alguna mejora en la detección de personajes para intentar reducir más el número de falsos positivos que pueda dar la aplicación, a pesar de que haya herramientas para poder juntar y borrar personajes, sería óptimo que el usuario tuviera que borrar y juntar el menor número de personajes.

Detección de etnia y sexo

Aunque ahora el algoritmo permite obtener la predicción de la etnia y el sexo, se sabe que Ethnea y Genni también obtienen en la etnia un porcentaje, por ejemplo, 80% NORDIC y 20% HISPANIC, sería bueno poder añadir como atributo todas las posibilidades de etnia que haya, aunque solo se muestre la principal.

Solución problema móvil

Aunque en el móvil se puede llegar a extraer el informe de características de la red, la visualización no es óptima, cuando un usuario intenta mover la red, esta desaparece y deja de visualizarse. Se investigó el problema, pero al final revisando el repositorio oficial donde se encuentra la librería empleada para la visualización, se observó que el problema también existía. Por tanto, se concluyó que no era un problema de implementación sino de la librería. Como cambio a futuro, se podría buscar alguna alternativa similar a esta visualización que no tenga este problema.

Bibliografía

1. Teoría de grafos - Wikipedia, la enciclopedia libre [Internet]. [citado 30 de enero de 2020]. Disponible en: https://es.wikipedia.org/wiki/Teor%C3%ADa_de_grafos#Aplicaciones

2. Ciencia de redes - Wikipedia, la enciclopedia libre [Internet]. [citado 30 de enero de 2020]. Disponible en: https://es.wikipedia.org/wiki/Ciencia_de_redes

3. Eigenvector Centrality - an overview | ScienceDirect Topics [Internet]. [citado 29 de enero de 2020]. Disponible en: https://www.sciencedirect.com/topics/computer-science/eigenvector-centrality

4. Las elites empresariales en América Latina, ¿está unidas?Networks provide happiness [Internet]. [citado 30 de enero de 2020]. Disponible en: http://networksprovidehappiness.com/las-elites-empresariales-en-america-latina-estan-unidas/

5. PageRank. En: Wikipedia, la enciclopedia libre [Internet]. 2020 [citado 29 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=PageRank&oldid=122982753

6. Louvain modularity. En: Wikipedia [Internet]. 2020 [citado 29 de enero de 2020]. Disponible en: https://en.wikipedia.org/w/index.php?title=Louvain_modularity&oldid=935297836

7. Clauset A, Newman MEJ, Moore C. Finding community structure in very large networks. Phys Rev E. 6 de diciembre de 2004;70(6):066111.

8. Método de Percolación de Cliques. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 29 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=M%C3%A9todo_de_Percolaci%C3%B3n_de_Cliques&oldid=119733426

9. Torvik V, Agarwal S. Ethnea --an instance-based ethnicity classifier based on geo-coded author names in a large-scale bibliographic database. En 2016.

10. Palacio J, Ruata C. Scrum Manager: Proyectos – Formación. 2008. 113 p.

11. EPUB. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 29 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=EPUB&oldid=122004791

12. Ramos H, Antonio E. Redes Complejas aplicado al análisis de la dinámica de Sistemas Sociales. Univ Nac Jorge Basadre Grohmann [Internet]. 2017 [citado 29 de enero de 2020]; Disponible en: http://repositorio.unjbg.edu.pe/handle/UNJBG/1580

13. García OC. Redes y Sistemas Complejos Cuarto Curso del Grado en Ingeniería Informática. :71.

14. Introducción a las redes complejas - Fernando Sancho Caparrini [Internet]. [citado 29 de enero de 2020]. Disponible en: http://www.cs.us.es/~fsancho/?e=80

15. Alejandro VÁO, Norman AG. Ejemplos prácticos con UCINET 6.85 y NETDRAW 1.48. :49.

16. Volumen+I%2FP1-1-GRAFOSCONCEPTOSBASICOS.pdf [Internet]. [citado 29 de enero de 2020]. Disponible en: http://bibing.us.es/proyectos/abreproy/11749/fichero/Volumen+I%252FP1-1-GRAFOSCONCEPTOSBASICOS.pdf

17. Coeficiente de agrupamiento. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 29 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Coeficiente_de_agrupamiento&oldid=117954147

18. Sharma D, Surolia A. Degree Centrality. En: Dubitzky W, Wolkenhauer O, Cho K-H, Yokota H, editores. Encyclopedia of Systems Biology [Internet]. New York, NY: Springer; 2013 [citado 29 de enero de 2020]. p. 558-558. Disponible en: https://doi.org/10.1007/978-1-4419-9863-7_935

19. Closeness Centrality - an overview | ScienceDirect Topics [Internet]. [citado 29 de enero de 2020]. Disponible en: https://www.sciencedirect.com/topics/computer-science/closeness-centrality

20. Betweenness Centrality - an overview | ScienceDirect Topics [Internet]. [citado 29 de enero de 2020]. Disponible en: https://www.sciencedirect.com/topics/computer-science/betweenness-centrality

21. Newman MEJ. A measure of betweenness centrality based on random walks. Soc Netw. 1 de enero de 2005;27(1):39-54.

22. Clique. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 29 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Clique&oldid=117943355

23. Networks -> Subgroups -> N-Cliques [Internet]. [citado 29 de enero de 2020]. Disponible en: http://www.analytictech.com/ucinet/help/attepv.htm

24. Networks -> Subgroups -> N-Clan [Internet]. [citado 29 de enero de 2020]. Disponible en: http://www.analytictech.com/ucinet/help/13_t_v_.htm

25. Networks -> Subgroups -> K-Plex [Internet]. [citado 29 de enero de 2020]. Disponible en: http://www.analytictech.com/ucinet/help/1pdb_fw.htm

26. Degeneracy (graph theory). En: Wikipedia [Internet]. 2019 [citado 29 de enero de 2020]. Disponible en: https://en.wikipedia.org/w/index.php?title=Degeneracy_(graph_theory)&oldid=931536671

27. Girvan–Newman algorithm - Wikipedia [Internet]. [citado 3 de febrero de 2020]. Disponible en: https://en.wikipedia.org/wiki/Girvan%E2%80%93Newman_algorithm

28. Guimerà R, Amaral LAN. Cartography of complex networks: modules and universal roles. J Stat Mech Online. 1 de febrero de 2005;2005(P02001):P02001-1-P02001-13.

29. Guimerà R, Sales-Pardo M, Amaral LAN. Classes of complex networks defined by role-to-role connectivity profiles. Nat Phys. enero de 2007;3(1):63-9.

30. Web scraping. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 29 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Web_scraping&oldid=118159308

31. Git. En: Wikipedia, la enciclopedia libre [Internet]. 2020 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Git&oldid=122867360

32. GitHub. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=GitHub&oldid=121159620

33. ZenHub - Agile Project Management for GitHub [Internet]. ZenHub. [citado 28 de enero de 2020]. Disponible en: https://www.zenhub.com

34. Free Git GUI Client - Windows, Mac & Linux | GitKraken [Internet]. GitKraken.com. [citado 28 de enero de 2020]. Disponible en: https://www.gitkraken.com/

35. GitHub Desktop [Internet]. GitHub Desktop. [citado 28 de enero de 2020]. Disponible en: https://desktop.github.com/

36. beautifulsoup4: Screen-scraping library.

37. urllib — URL handling modules — Python 3.8.1 documentation [Internet]. [citado 28 de enero de 2020]. Disponible en: https://docs.python.org/3/library/urllib.html

38. Ethnea [Internet]. [citado 28 de enero de 2020]. Disponible en: http://abel.lis.illinois.edu/cgi-bin/ethnea/search.py

39. NetworkX. En: Wikipedia [Internet]. 2019 [citado 28 de enero de 2020]. Disponible en: https://en.wikipedia.org/w/index.php?title=NetworkX&oldid=915305818

40. Community detection for NetworkX's documentation — Community detection for NetworkX 2 documentation [Internet]. [citado 28 de enero de 2020]. Disponible en: https://python-louvain.readthedocs.io/en/latest/

41. Aynaud T. taynaud/python-louvain [Internet]. 2020 [citado 28 de enero de 2020]. Disponible en: https://github.com/taynaud/python-louvain

42. Aslak U. ulfaslak/network_styling_with_d3 [Internet]. 2020 [citado 28 de enero de 2020]. Disponible en: https://github.com/ulfaslak/network_styling_with_d3

43. Aslak U, Maier B. Netwulf: Interactive visualization of networks in Python. J Open Source Softw. 1 de octubre de 2019;4(42):1425.

44. Maier BF. benmaier/netwulf [Internet]. 2020 [citado 28 de enero de 2020]. Disponible en: https://github.com/benmaier/netwulf

45. Flask. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Flask&oldid=119183568

46. Welcome to Flask — Flask 1.0.2 documentation [Internet]. [citado 10 de junio de 2019]. Disponible en: http://flask.pocoo.org/docs/1.0/

47. Flask-Babel — Flask Babel 1.0 documentation [Internet]. [citado 28 de enero de 2020]. Disponible en: https://pythonhosted.org/Flask-Babel/

48. Bootstrap (framework). En: Wikipedia, la enciclopedia libre [Internet]. 2020 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Bootstrap_(framework)&oldid=122653369

49. contributors MO Jacob Thornton, and Bootstrap. Introduction [Internet]. [citado 28 de enero de 2020]. Disponible en: https://getbootstrap.com/docs/4.0/getting-started/introduction/

50. JavaScript. En: Wikipedia, la enciclopedia libre [Internet]. 2020 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=JavaScript&oldid=123113801

51. AJAX. En: Wikipedia, la enciclopedia libre [Internet]. 2020 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=AJAX&oldid=122422989

52. jQuery. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=JQuery&oldid=120193158

53. Wikidot - Free and Pro Wiki Hosting [Internet]. [citado 28 de enero de 2020]. Disponible en: https://www.wikidot.com/

54. MediaWiki [Internet]. [citado 28 de enero de 2020]. Disponible en: https://www.mediawiki.org/wiki/MediaWiki

55. Heroku. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Heroku&oldid=122159086

56. OpenShift. En: Wikipedia, la enciclopedia libre [Internet]. 2020 [citado 28 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=OpenShift&oldid=122792474

57. Poedit. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 29 de enero de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Poedit&oldid=117398891

58. Crea y comparte diseños, maquetas y prototipos rápidamente | Adobe XD [Internet]. [citado 5 de febrero de 2020]. Disponible en: https://www.adobe.com/es/products/xd/details.html

59. (17) Flask Tutorial Web Development with Python 1 - Intro - YouTube [Internet]. [citado 28 de enero de 2020]. Disponible en: https://www.youtube.com/watch?v=Lv1fv-HmkQo

60. Sivji A. Building a Flask Web Application (Flask Part 2) [Internet]. [citado 28 de enero de 2020]. Disponible en: ./flask-part2-building-a-flask-web-application.html

61. Modelo–vista–controlador - Wikipedia, la enciclopedia libre [Internet]. [citado 3 de febrero de 2020]. Disponible en: https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador

62. ICEUTE 2020 – BURGOS (SPAIN) – JUNE [Internet]. [citado 5 de febrero de 2020]. Disponible en: http://2020.iceuteconference.eu/

63. Arce LMC. lca0037/GII18.0U-Ububooknet [Internet]. 2019 [citado 5 de febrero de 2020]. Disponible en: https://github.com/lca0037/GII18.0U-Ububooknet

64. Beveridge A, Shan J. Network of Thrones. Math Horiz. 1 de abril de 2016;23(4):18-22.

65. Who Is the Most Important Character in Frozen? What Networks Can Tell Us About the World [Internet]. Frontiers for Young Minds. [citado 5 de febrero de 2020]. Disponible en: https://kids.frontiersin.org/article/10.3389/frym.2019.00099

66. Dekker N, Kuhn T, van Erp M. Evaluating named entity recognition tools for extracting social networks from novels. PeerJ Comput Sci. 18 de abril de 2019;5:e189.