

**APLICACIÓN DE TÉCNICAS DE ANÁLISIS  
DE REDES SOCIALES EN EL SISTEMA  
BIKE-SHARING DE BiciMAD.**

DATALAB CIENCIAS SOCIALES.

Curso 2021/2022

FACULTAD DE CIENCIAS.

MÁSTER EN CIENCIA DE DATOS.

Autor: Jesús Octavio Raboso.

Santander, 22 de abril de 2022.



## Resumen

Los sistemas de bike-sharing han sido objeto de interés en los últimos tiempos debido a su potencial para el desarrollo de ciudades inteligentes y ecológicas. Identificar las dinámicas de estos sistemas permite estudiar qué decisiones tendrán más impacto y fomentarán el uso de la bicicleta. En concreto, se aplican las técnicas del Análisis de Redes Sociales para el servicio de bike-sharing de Madrid, BiciMAD. Se considera la red generada a partir de datos de junio de 2019. Para realizar el análisis, se emplea el software open-source Gephi y la librería *networkx* de Python. Se ofrece una propuesta para almacenar los datos en Neo4j, un software libre de bases de datos orientado a grafos.

**Palabras clave:** *Análisis de Redes Sociales, BiciMAD, medidas de centralidad, Neo4j.*



# Índice general

<b>Resumen.</b>	<b>III</b>
<b>Índice general</b>	<b>v</b>
<b>Introducción.</b>	<b>1</b>
<b>1. Análisis de Redes Sociales.</b>	<b>3</b>
1. Estructura de los datos. . . . .	3
2. Potencial de los datos. . . . .	5
3. Definición y análisis de la red social. . . . .	6
<b>2. Neo4J.</b>	<b>27</b>
1. Construcción de la red social. . . . .	27
2. Consultas propuestas. . . . .	28
<b>3. Conclusiones y trabajo futuro.</b>	<b>35</b>
<b>Anexo</b>	<b>36</b>
<b>Bibliografía.</b>	<b>41</b>



# Introducción.

La bicicleta constituye un medio de transporte cómodo, saludable y, en zonas congestionadas, una alternativa más rápida. Por ello, los sistemas de bicicletas compartidas -bike-sharing- ponen a disposición del ciudadano una serie de estaciones en las que se puede coger o dejar una bicicleta. Normalmente, los usuarios se identifican mediante algún código digital y liberan una bicicleta que puede ser devuelta en cualquier otro punto.

La primera iniciativa de bicicletas públicas compartidas en el mundo fue *Whitte Fietsen*, lanzada en Ámsterdam en 1964. Consistía en unas bicicletas pintadas de blanco y distribuidas por toda la ciudad, sin candados y libres para que cualquier persona las utilizase. Tras un mes, la mayoría de las bicicletas habían desaparecido. Aunque fuese un experimento fallido y limitado -sólo 10 bicicletas se pusieron a disposición de los usuarios- asentó las bases para este concepto de movilidad.

La segunda generación de estos servicios arrancó en Copenhague en 1995. Esta vez, sí era necesario el previo pago para usar las bicicletas.

Desde entonces, gran parte de las ciudades del mundo han adoptado este sistema de movilidad. La primera en hacerlo en España fue Vitoria, en 2004. Si bien, nos centraremos en el servicio de bicicletas de alquiler de la ciudad de Madrid, BiciMAD.

Al igual que otros servicios similares, los objetivos de BiciMAD son proporcionar un elemento alternativo de transporte limpio y saludable para el ciudadano y fomentar el uso de la bicicleta en la ciudad. A día de hoy, el sistema está compuesto por:

- 2964 bicicletas,
- 6315 anclajes,
- 264 estaciones.

Cada estación está compuesta por un conjunto de anclajes -bases- en las que se deposita o extrae la bicicleta utilizada.

BiciMAD destaca por el empleo de las Tecnologías de la Información y la Comunicación para mejorar la experiencia del usuario, aumentar la interactividad multiplataforma y dar acceso a la información a través de la web y dispositivos móviles.

Los datos publicados por BiciMAD pueden obtenerse en el [Portal de Datos Abiertos del Ayuntamiento de Madrid](#), que también ofrece datasets relativos a actividades de MercaMadrid, datos de tráfico, meteorológicos, demográficos... En definitiva, datos interesantes con los que pueden cruzarse los de BiciMAD y, así, extraer conclusiones más avezadas. En concreto, los datos relativos a BiciMAD pueden consultarse en el siguiente [enlace](#).

El avance de las tecnologías GIS y la compartición de datasets permiten que el Análisis de Redes Sociales y otros conceptos de la Teoría de Grafos sean aplicados en el campo de la movilidad urbanística.

En el contexto que nos atañe, los términos *red* o *grafo* hacen referencia al conjunto de vértices o nodos -estaciones de bicicletas- y rutas o arcos -viajes entre dos estaciones- que componen el sistema.

En general, las redes de transporte describen infraestructuras estáticas y permanentes -carreteras, vías ferroviarias...- con horarios prefijados. Sin embargo, los modos de transporte individuales muestran rutas flexibles, cambiantes y horarios diversos. Dada la apuesta por la transparencia, en BiciMAD se registra cada movimiento de cada usuario, luego un análisis exhaustivo de la red social que determina el sistema puede ayudar a la planificación urbanística de la ciudad, a ver dónde son necesarias más bicicletas y a comprobar, cruzando con otros datos, si el uso de la bicicleta está condicionado por indicadores sociales, económicos, meteorológicos...

En un [estudio previo](#), se consideraron los datos más actuales en aquel momento -datos entre julio de 2018 y junio de 2019- y se estableció junio de 2019 como el mes más interesante. Analizaremos la red social que los usuarios conformaron el dicho mes.

# Capítulo 1

## Análisis de Redes Sociales.

En las próximas secciones, se detalla la estructura de los datos que ofrece BiciMAD y cómo se ha creado la red. Además, se calculan las principales medidas de centralidad de la red generada. Se abordarán otras cuestiones interesantes como el potencial de los datos o diversas formas de representación visual de la red.

### 1. Estructura de los datos.

Los datos que ofrece por BiciMAD se encuentran disponibles en el siguiente [enlace](#). El volumen de cada dataset es bastante grande, de modo que emplearemos [Apache Spark](#) para su tratamiento como archivos RDD.

Los archivos RDD -Resilient Distributed Datasets- tienen como características fundamentales:

- Son la principal abstracción de datos de Apache Spark.
- Están particionados en los nodos del clúster. Las RDDs se encuentran distribuidas entre las máquinas de Apache Spark, es decir, se almacenen en distintas ubicaciones, luego es más tolerante a fallos.
- Usan evaluación perezosa.

Todos los datasets de BiciMAD se almacenan en archivos .json. Cada entrada posee los siguientes campos:

- **user\_day\_code**: código único que BiciMAD asigna a cada usuario en un día determinado.
- **idplug\_base**: base de la estación de destino en la que la bicicleta es depositada.
- **user\_type**: tipo de usuario. Existen 4 tipos de usuario identificados con números enteros:

- 0: tipo de usuario no identificado.
- 1: usuario con abono anual.
- 2: usuario ocasional.
- 3: trabajador de BiciMAD.

- **idunplug\_base**: base de la estación de origen de la cual la bicicleta es extraída.
- **travel\_time**: duración del viaje en segundos.
- **idunplug\_station**: estación de destino.
- **ageRange**: rango de edad en el que se encuentra el usuario. Puede tomar los siguientes valores:
  - 0: edad del usuario no identificada.
  - 1; 0-16 años.
  - 2; 17-18 años.
  - 3; 19-26 años.
  - 4; 27-40 años.
  - 5; 41-65 años.
  - 6; 66 años o más.
- **idplug\_station**: estación de destino.
- **unplug\_hourTime**: hora de inicio del trayecto.
- **zip\_code**: código postal donde, se supone, reside el usuario.

También leeremos otros archivos que contienen información geográfica sobre las estaciones, en concreto:

- **Número**: número identificador de la estación.
- **Gis\_X**: coordenada *x* en GIS.
- **Gis\_Y**: coordenada *y* en GIS.
- **Fecha de Alta**: fecha en la que la estación se incorpora al servicio de BiciMAD.
- **Distrito**: distrito en el que se encuentra la estación.
- **Barrio**: barrio en el que se encuentra la estación.
- **Calle**: calle en la que se encuentra la estación.
- **Nº Finca**: número de Finca.
- **Tipo de reserva**.
- **Plazas**: número de anclajes o bases en la estación.

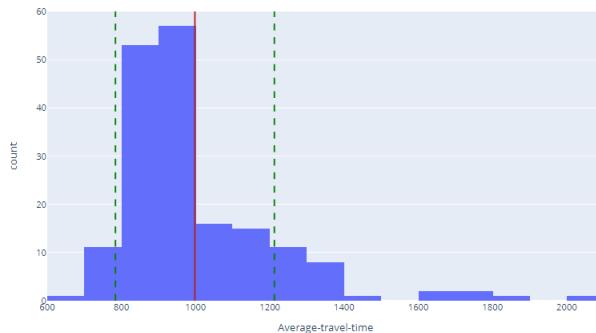


Figura 1.1: Histograma para la duración media de los viajes que parten de cada estación.

- **LONGITUD:** longitud en la que se encuentra la estación.
- **LATITUD:** latitud en la que se encuentra la estación.
- **DIRECCIÓN:** dirección completa en la que se encuentra la estación, no únicamente la calle sino también el número.

La definición y estructura detallada de los datos puede consultarse en el siguiente [enlace](#).

El tratamiento detallado de los datos puede consultarse en el notebook disponibles en el siguiente [repositorio de GitHub](#).

## 2. Potencial de los datos.

Antes de definir la red que conforma el sistema, se pretende dar una idea del gran potencial de los datos disponibles, sobre todo si se cruzan con otros datasets.

Dado el conjunto de datos, seleccionaremos aquellos que duran más de 30 segundos y cuyas estaciones de origen y destino sean distintas. Deben darse ambas condiciones, pues viajes que empiezan y acaban en la misma estación y duran menos de 30 segundos se consideran fallidos.

### Duración media de los viajes que parten de cada estación.

Se calcula la duración media de los viajes que parten de cada estación teniendo en cuenta que el tiempo se guarda en segundos.

Puede verse su distribución mediante un histograma (Fig 1.1). La media de duración de los viajes que parten de las estaciones es de 997 segundos -16 minutos- y la desviación típica es de 214 segundos -3.5 minutos-. Se corresponden con trayectos de poca distancia que, a priori, no requieren un gran esfuerzo para las personas, rutas rápidas que en otro medio de transporte quizás se prolongarían.

### **Estaciones de destino.**

Para cada estación, hallamos las 5 estaciones más frecuentes a las que se llega desde un viaje iniciado en la estación considerada. Posteriormente, realizaremos esta misma operación aunque sin restringir en un ranking.

Por ejemplo, se observa que desde la estación identificada como 1, hay 116 viajes que parten hasta la estación 175; 93 viajes que llegan a la estación 41; 63 viajes que llegan a la estación 49; 60 viajes que llegan a la estación 57; 56 viajes que llegan a la estación 133.

### **Estaciones de origen.**

De forma análoga, para cada estación, hallamos las 5 estaciones más frecuentes a las que se llega desde un viaje iniciado en la estación considerada.

Por ejemplo, se observa que hasta la estación identificada como 1, hay 148 viajes que llegan desde la estación 175; 90 que llegan desde la estación 57; 80 que llegan desde la estación 41; 73 que llegan desde la estación 49; 61 que llegan desde la estación 129.

Teniendo en cuenta las estaciones de destino anteriormente calculadas, parece que existe una clara relación entre las estaciones 1 y 175.

### **Visualización.**

Estos datos pueden representarse sobre el mapa de la ciudad de Madrid. Cada estación se corresponde con un pop-up que, al desplegarlo, muestra la información de los problemas antes considerados. También se muestra un círculo centrado en cada estación cuyo radio es proporcional a la duración media de los viajes que parten de la estación (Fig 1.2).

## **3. Definición y análisis de la red social.**

Los datos ofrecidos por BiciMAD no se encuentran en el formato adecuado para establecer una red social, de modo que hay que hacer ciertas transformaciones y cruzarlos con un dataset de tipo GIS que contiene la localización espacial de las estaciones, su dirección, barrio y otros datos de interés. Aunque en este informe nos dedicaremos únicamente a mostrar los resultados del Análisis de Redes Sociales y su interpretación, tanto el preprocesado de los datos como los pasos para obtener los resultados pueden consultarse en el [repositorio de GitHub](#).

El resultado final es un DataFrame con la estructura de (Fig 1.3) donde cada columna significa:

- **Source:** estación de origen.
- **Target:** estación de destino.

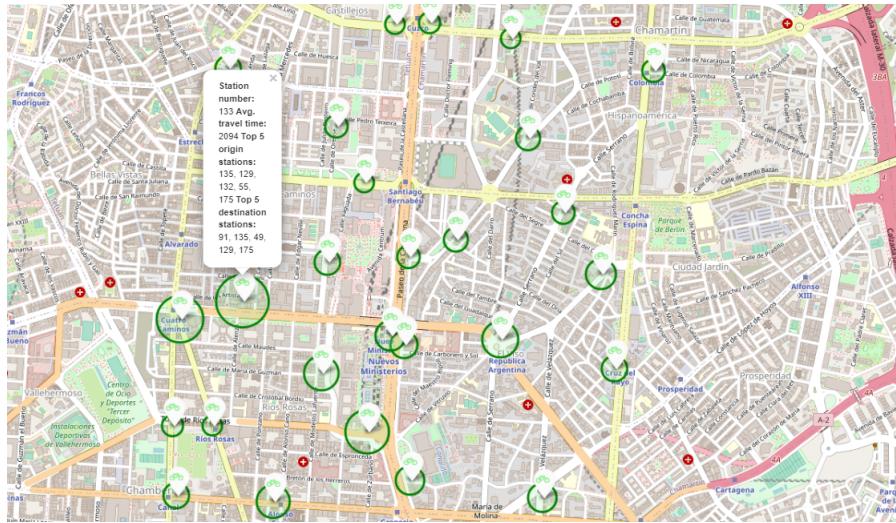


Figura 1.2: Mapa de las estaciones con los problemas de duración media, estaciones destino y estaciones origen planteados en la sección 2.

Source	Target	Weight	Source_Barrio	Source_Direccion	Source_Lat	Source_Lon	Target_Barrio	Target_Direccion	Target_Lat	Target_Lon	Weight_Inverse
0	1	175	116	01-06 SOL CALLE DE ALCALÁ, 2	40.4171107953151	-3.70199802576925	02-05 DELICIAS	CALLE DE JAIME EL CONQUISTADOR, 30	40.396332942252	-3.69830799524925	0.008621
0	1	41	93	01-06 SOL CALLE DE ALCALÁ, 2	40.4171107953151	-3.70199802576925	01-02 EMBAJADORES	CALLE DE ATOCHA, 56	40.412228999799	-3.6988977972563	0.010753
0	1	49	63	01-06 SOL CALLE DE ALCALÁ, 2	40.4171107953151	-3.70199802576925	01-02 EMBAJADORES	RONDA DE ATOCHA, 36	40.406049872348	-3.69926246482099	0.015873
0	1	57	60	01-06 SOL CALLE DE ALCALÁ, 2	40.4171107953151	-3.70199802576925	01-04 JUSTICIA	CALLE DE SANTA CRUZ DE MARCENAO, 24	40.429634201919	-3.71281811499328	0.016667
0	1	133	56	01-06 SOL CALLE DE ALCALÁ, 2	40.4171107953151	-3.70199802576925	07-05 RÍOS ROSAS	CALLE DE DULCINEA, 3	40.447167454003	-3.70002920994525	0.017657

Figura 1.3: DataFrame que define la red social de BiciMAD.

- **Weight:** número de viajes entre la estación de origen y la estación de destino,
- **Source \_ Barrio:** barrio o distrito al que pertenece la estación de origen.
- **Source \_ Dirección:** dirección de la estación de origen.
- **Source \_ Lat:** latitud de la estación de origen.
- **Source \_ Lon:** longitud de la estación de origen.
- **Target \_ Barrio:** barrio o distrito al que pertenece la estación de destino.
- **Target \_ Dirección:** dirección de la estación de destino.
- **Target \_ Lat:** latitud de la estación de destino.
- **Target \_ Lon:** longitud de la estación de destino.
- **Weight-Inverse:** inversa de las columnas Weight.

En la red subyacente al servicio de BiciMAD, los nodos se corresponden con estaciones y los arcos con los viajes entre las estaciones destino y origen. Su peso se corresponde con el número de viajes. Es decir, tenemos una red dirigida y valorada en la que los pesos tienen una connotación positiva en el sentido de que cuanto mayor sea el peso, más *unidas* se encuentran las estaciones involucradas en el arco. Por ello y para calcular algunas métricas, ha de emplearse otra valoración como, por ejemplo, la inversa de los pesos. En su momento, se abordará este problema.

Si se dibuja la red, se obtiene el resultado mostrado en Fig 1.4. Es inmanejable y apenas puede extraerse ninguna conclusión, de modo que se ha optado por realizar dos funciones que reciben como input un nodo y dibujan los subgrafos que tienen a dicho nodo como origen como destino (Fig 1.5).

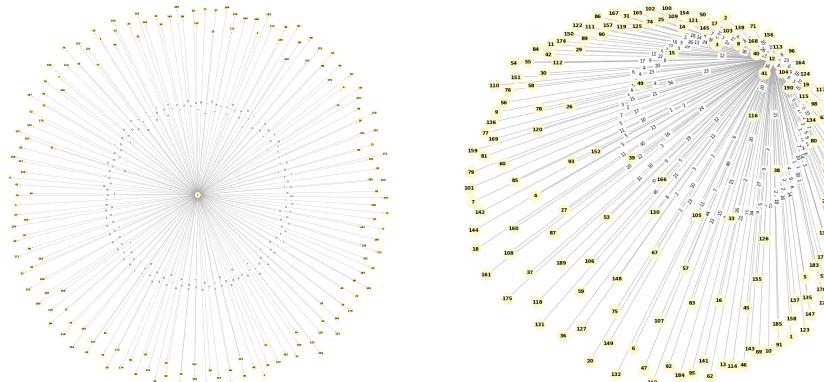


Figura 1.5: Subgrafos que muestran el nodo de la estación 1 como origen y como destino.

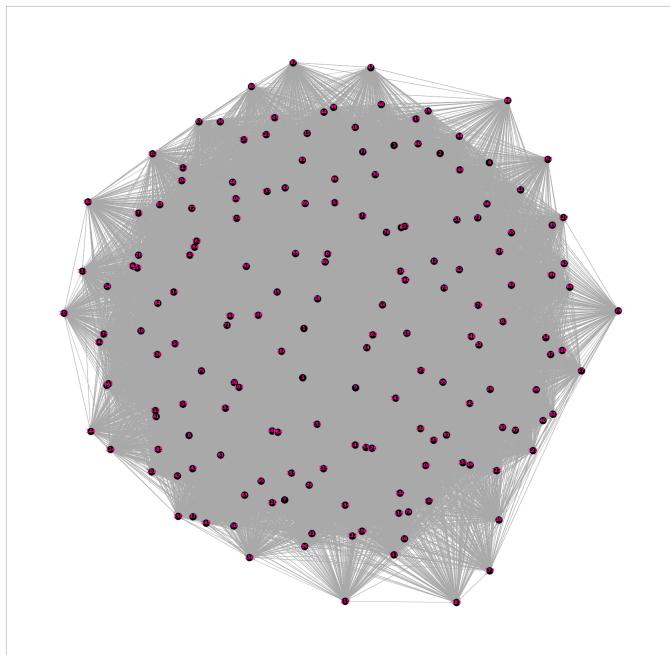


Figura 1.4: Visualización social de BiciMAD.

Una alternativa de representación puede ser un heatmap (Fig 1.6). En el eje vertical, colocamos los nodos origen y en el horizontal, colocamos los nodos destino. Los puntos de corte los coloreamos en función del número de viajes entre el origen y el destino.

### Análisis de medidas de centralidad.

En lo que sigue, se analizan las principales medidas de centralidad para cuantificar la importancia de los nodos y estudiar la estructura de la red. Se han empleado la librería `networkx` y el software Gephi.

Se recomienda seguirlo junto al notebook `bicimad.ipynb`, que reproduce los resultados en detalle y permite ver las imágenes con una resolución adecuada. Por comodidad, se ha hecho en Python, si bien puede seguirse la reproducción parcial en el archivo `bicimad.gephi`.

#### Tamaño.

El tamaño de un grafo  $G = (V, E)$  donde  $V$  es el conjunto de nodos y  $E$  es el conjunto de arcos se refiere al cardinal de dichos conjuntos.

En nuestra red, existen 172 nodos -estaciones- y 27011 arcos. El número de arcos permite atisbar ciertas ideas sobre la conexión de la red. Si cada nodo estuviese conectado con los restantes, habría un total de  $172 \cdot 171 = 29412$  arcos, de modo que puede decirse que ha habido viajes entre casi todos los pares de estaciones.

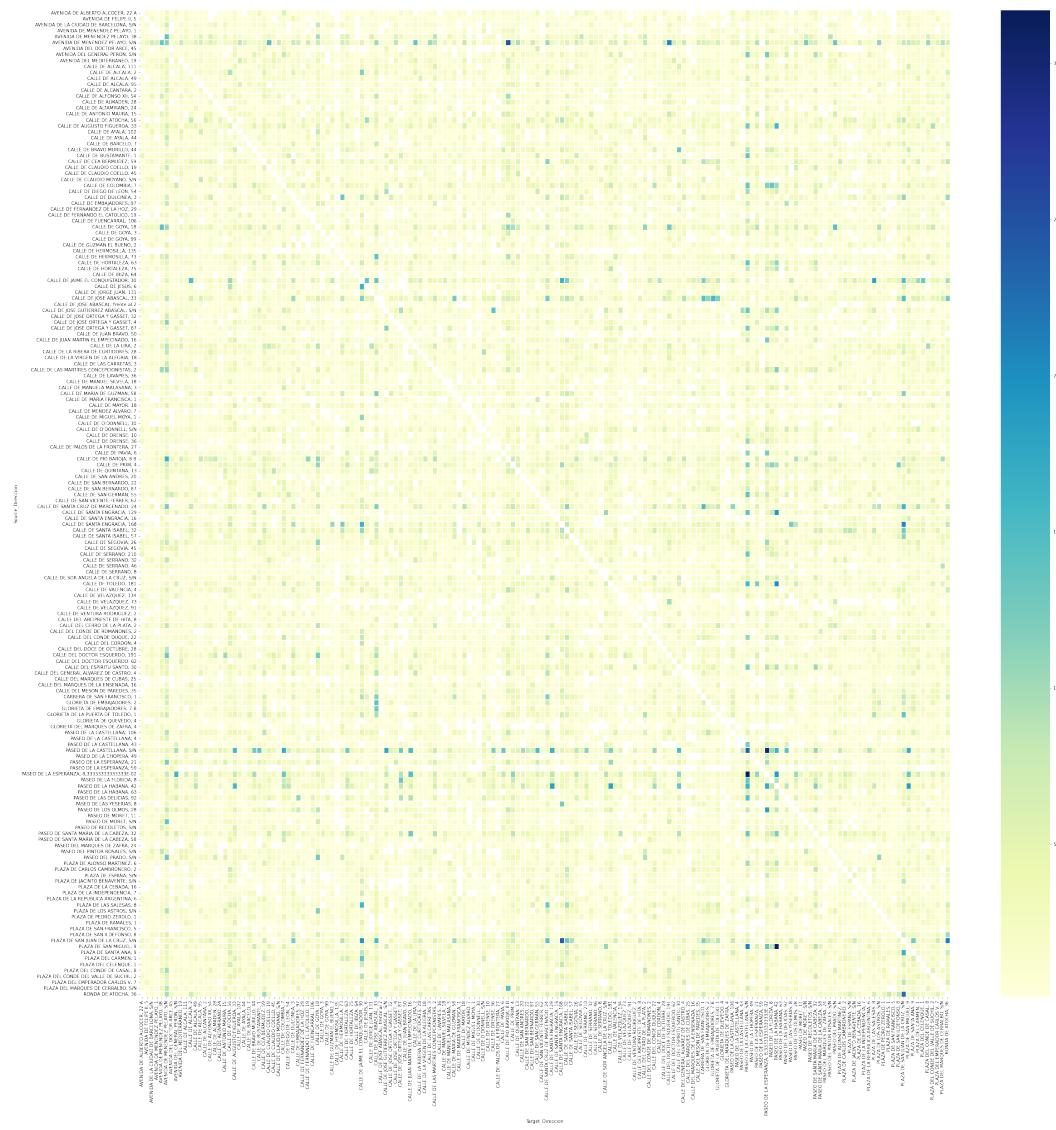


Figura 1.6: Visualizacion de la red como heatmap.

Para calcular el número de nodos se ha empleado la función `number_of_nodes()` y para calcular el número de arcos se ha empleado la función `number_of_edges()`.

### Conexión.

Se dice que un grafo dirigido es *débilmente conexo* si el grafo es conexo cuando la dirección de los arcos es ignorada. La red de BiciMAD red débilmente conexa, algo que puede comprobarse haciendo uso de la función `is_weakly_connected()`.

Se dice que un grafo dirigido es *fuertemente conexo* si para cada par de vértices  $u, v \in V$  existe un camino desde  $u$  hasta  $v$  y un camino desde  $v$  hasta  $u$  teniendo en cuenta el sentido de los arcos. La red de BiciMAD es fuertemente conexa, algo que puede comprobarse haciendo uso de la función `is_strongly_connected()`.

### Diámetro.

El diámetro de un grafo  $G = (V, E)$  se define como la máxima excentricidad, es decir:

$$\max\{d(u, v) | u, v \in V\} \quad (3.1)$$

Intuitivamente, se corresponde con el máximo de las distancias entre todos los pares de nodos.

Tanto Gephi como la librería networkx de Python establece que el diámetro de la red es 2. ¿Tiene sentido esta métrica? Al menos, Python no permite especificar cuál es el peso de los arcos y Gephi no detalla cómo calcula el diámetro. ¿Se asume que todos los arcos están valorados como 1? Esta métrica no debe ser considerada.

### Degree.

El *degree* de un nodo es el número de conexiones que posee dicho nodo. Si consideramos la red de BiciMAD como no valorada, se corresponde con el número de arcos -de salida y entrada- que confluyen -salen o entran- en un nodo. Cada arco tiene peso 1. Se emplea `degree()` para calcular el grado de cada nodo con el argumento `weights = None` para que no tenga en cuenta los pesos.

La distribución del degree puede verse en el histograma Fig 1.7. Observamos que la mayoría de los nodos se acumulan en el último bin, algo que concuerda con lo visto hasta ahora en las secciones de tamaño y conexión. Intuitivamente, esto quiere decir que gran parte de los nodos tienen viajes que llegan del resto y tienen viajes que parten hacia el resto.

Podemos visualizar esta medida sobre un grafo. En Fig 1.8, cuanto mayor es el tamaño y más oscuro es el color de los nodos, mayor es su degree.

Si bien, esta medida no es del todo descriptiva para nuestra red. Puede calcularse el *weighted degree* o *grado valorado* empleando el argumento `weights = "Weights"` en la función `degree()`. En Fig 1.9 se observa el cambio en la distribución del weighted degree con respecto al degree.

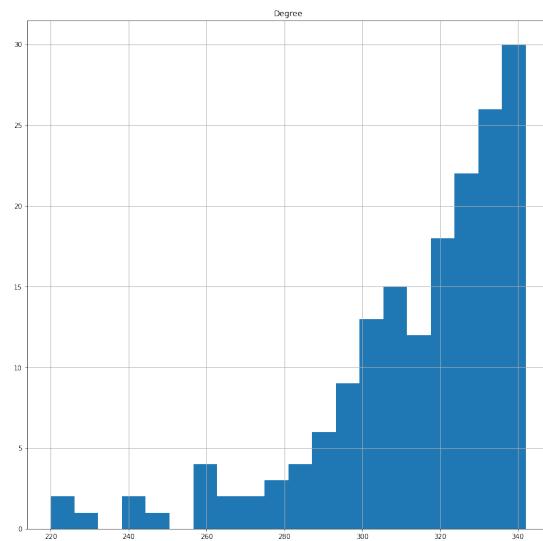


Figura 1.7: Distribución del degree considerando la red como no valorada.

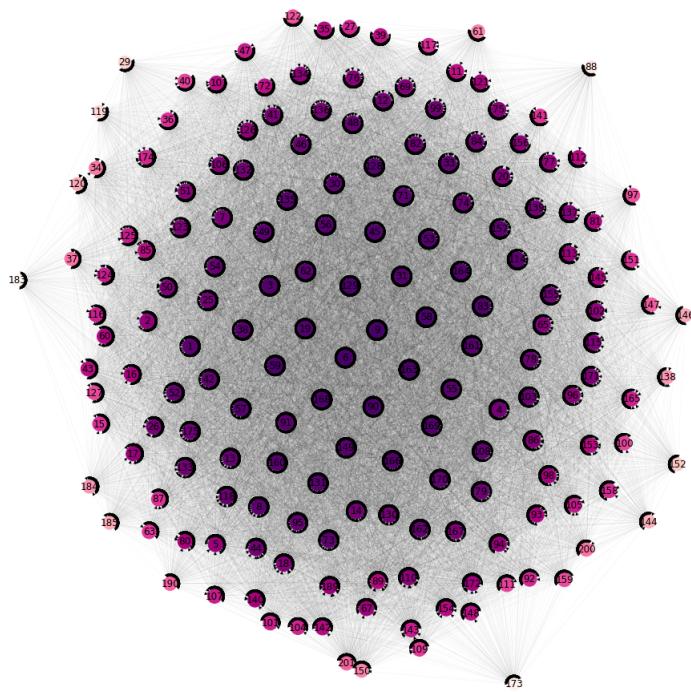


Figura 1.8: Visualización del degree considerando la red como dirigida y no valorada.

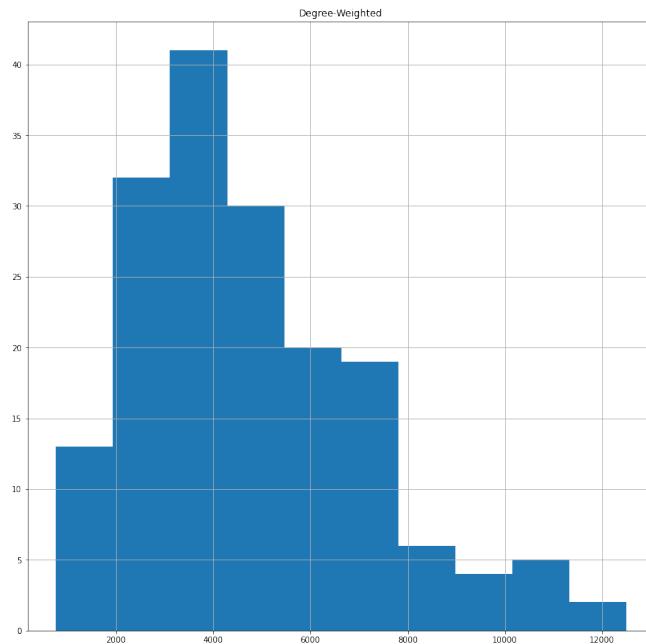


Figura 1.9: Histograma para el weighted degree.

De forma análoga, puede definirse el *in-degree* como el número de arcos que desembocan en un nodo al considerar la red como no valorada y el *weighted in-degree* como el total de viajes que desembocan en un nodo al considerar la red como valorada.

Nuevamente, la distribución de estas dos métricas difiere al considerar la red como no valorada o valorada (Fig 1.10).

Las 2 estaciones con menor weighted in-degree, es decir, aquellas a las que menos viajes llegan se corresponden con:

- Estación 183 con un total de 401 viajes. Situada en la Calle de Hermosilla, en el barrio Fuente del Berro, se enmarca en el distrito de Salamanca. Se trata de una zona residencial pero, guiándonos por estereotipos, probablemente sus habitantes prefieran un transporte distinto a la bicicleta para volver a casa. La estación se halla cerca del WiZink Center, pero, la temporada deportiva ha llegado a su fin en junio -mes de nuestro análisis-. Quizá en otros meses la afluencia en dicha estación es superior.
- Estación 29 con un total de 643 viajes. Situada en la Calle del Marqués de Cubas, en el barrio de Palacio, está cerca de la Fuente de Neptuno y el Museo del Prado. A pesar de ser una zona turística, parece que no se emplea mucho dicha estación como destino. Lo mismo ocurre con otras estaciones de poco weighted in-degree. Quizá esto sea sinónimo de que quienes realmente utilizan BiciMAD no son turistas sino residentes que toman la bicicleta para sus trayectos habituales.

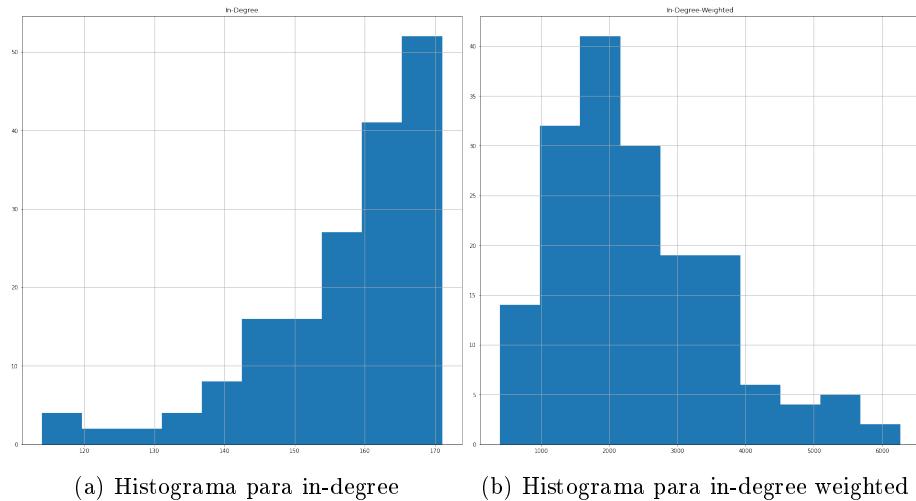


Figura 1.10: Comparación de histogramas para el in-degree y el in-degree weighted

Del mismo modo, las 2 estaciones con mayor weighted in-degree, es decir, aquellas a las que más viajes llegan, se corresponden con:

- Estación 136 con un total de 6265 viajes. Situada en Paseo de la Esperanza, en el barrio de Delicias, se trata de una zona cercana a Madrid Río y próxima a la multitudinaria estación de Embajadores. Dada su localización, es comprensible su gran afluencia.
- Estación 129 con un total de 5805 viajes. Situada en la Calle de José Abascal, en el barrio de Ríos Rosas, se trata de una zona céntrica, cercana al hospital Gregorio Marañón y próxima a varias estaciones de metro.

En Fig 1.11a, cuanto mayor es el tamaño y más oscuro es el nodo, mayor es el in-degree weighted. En Fig 1.11b se han representado en rojo las estaciones cuyo in-degree weighted está por debajo del percentil 10 y, en verde, aquellas cuyo in-degree weighted está por encima del percentil 90. No se observa ningún patrón espacial más allá de que las estaciones de barrios de gran poder adquisitivo como Salamanca y Retiro están por debajo del percentil 10.

Puede repetirse este mismo ejercicio con el *out-degree*, que se define como el número de arcos que parten de un nodo considerando la red como no valorada y el *weighted out-degree* que se define como el total de viajes que parten de un nodo considerando la red como valorada.

Nuevamente, la distribución de estas dos métricas difieren al considerar la red como no valorada o valorada (Fig 1.12)

Las 2 estaciones con menor weighted out-degree, es decir, aquellas de las que menos viajes parten se corresponden con:

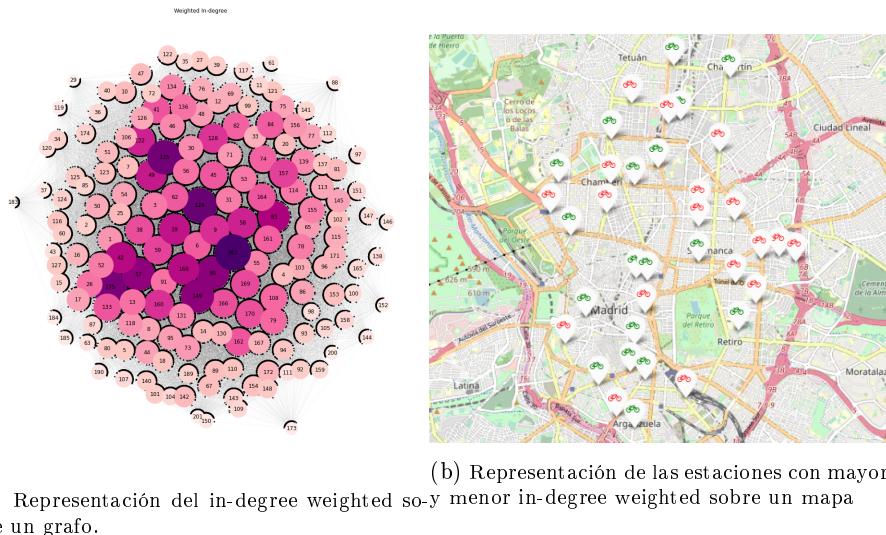


Figura 1.11: Visualizaciones para el weighted in-degree.

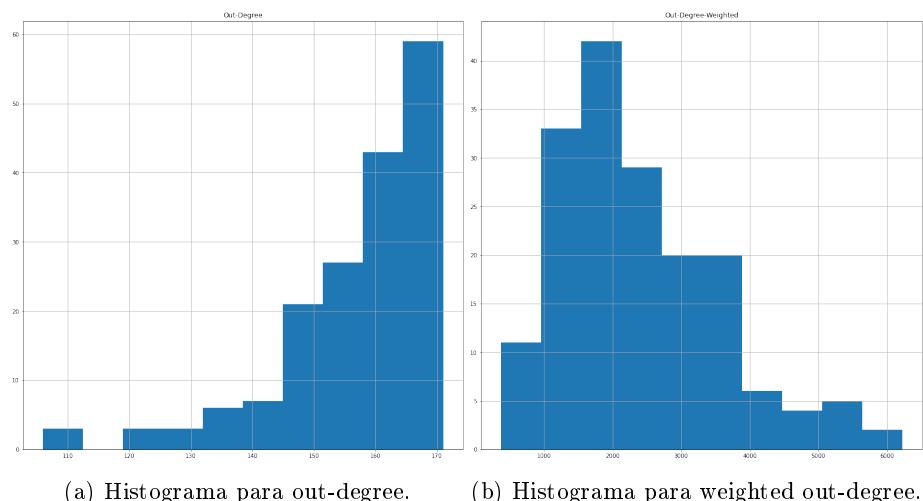


Figura 1.12: Comparación de histogramas para el in-degree y el in-degree weighted

- Estación 183 con un total de 375 viajes. Situada en la Calle de Hermosilla, en el barrio Fuente del Berro, se enmarca en el distrito de Salamanca. Se corresponde también con la estación a la que menos viajes llegan.
- Estación 29 con un total de 652 viajes. Situada en la Calle del Marqués de Cubas, en el barrio de Palacio, está cerca de la Fuente de Neptuno y el Museo del Prado. Se corresponde también con la segunda estación a la que menos viajes llegan.

Del mismo modo, las 2 estaciones con mayor weighted out-degree, es decir, aquellas a las que más viajes llegan, se corresponden con:

- Estación 136 con un total de 6223 viajes. Situada en Paseo de la Esperanza, en el barrio de Delicias. Se corresponde también con la estación a la que más viajes llegan.
- Estación 129 con un total de 5805 viajes. Situada en la Calle de José Abascal, en el barrio de Ríos Rosas. Se corresponde también con la segunda estación a la que más viajes llegan.

Este tipo de coincidencias nos hace pensar que los usuarios son clientes asiduos de BiciMAD y, probablemente, empleen la bicicleta como medio de transporte habitual.

En Fig 1.13a, cuanto mayor es el tamaño y más oscuro es el nodo, mayor es el weighted out-degree. En Fig 1.13b se han representado en rojo las estaciones cuyo out-degree weighted está por debajo del percentil 10 y, en verde, aquellas cuyo out-degree weighted está por encima del percentil 90. No se observa ningún patrón espacial más allá de que las estaciones de barrios de gran poder adquisitivo como Salamanca y Retiro están por debajo del percentil 10. La localización es similar a la del weighted in-degree.

Finalmente, estas 6 métricas se han unido en un único dataframe (Fig 1.14) y se han hallado sus valores medios.

- Average degree: 314.0813953488372.
- Average weighted degree: 4791.441860465116.
- Average in-degree: 157.0406976744186.
- Average weighted in-degree: 2395.720930232558.
- Average out-degree: 157.0406976744186.
- Average weighted out-degree: 2395.720930232558.

Las coincidencias no son casuales sino que están justificadas al ser la red dirigida y valorada.

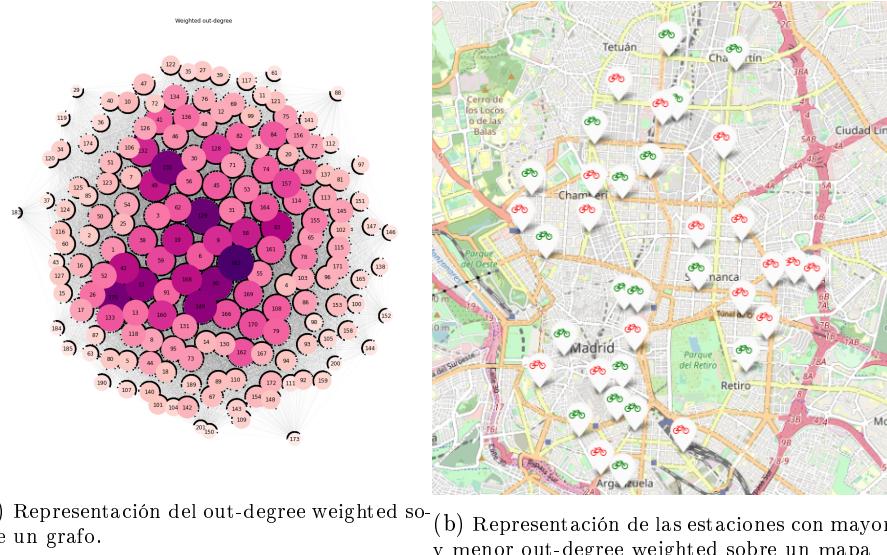


Figura 1.13: Visualizaciones para el eighted out-degree.

	<b>Id</b>	<b>Degree</b>	<b>Degree-Weighted</b>	<b>In-Degree</b>	<b>In-Degree-Weighted</b>	<b>Out-Degree</b>	<b>Out-Degree-Weighted</b>
0	1	335	5514	169	2746	166	2768
1	2	319	3229	159	1593	160	1636
2	3	338	6212	169	3118	169	3094
3	4	329	4027	165	2030	164	1997
4	5	317	3467	155	1770	162	1697
...	...	...	...	...	...	...	...
167	185	261	1799	133	898	128	901
168	189	317	3014	162	1505	155	1509
169	190	283	2300	139	1127	144	1173
170	200	278	1637	140	802	138	835
171	201	282	1621	146	824	136	797

Figura 1.14: Dataframe recopilando degree, degree weightes, in.....

### Closeness.

Sea el grafo  $G = (V, E)$  donde  $|V| = N$ . El closeness de un nodo  $u$  es el inverso de la suma de la longitud de los caminos más cortos desde  $u$  hasta los  $N - 1$  nodos restantes. Puesto que la suma de las distancias depende del número de nodos, se normaliza mediante:

$$C(u) == \frac{N - 1}{\sum_{v \in V} d(v, u)}$$

donde  $d(u, v)$  es la distancia del camino más corto entre los nodos  $v$  y  $u$ .

Se trata del inverso del farness e, intuitivamente, permite identificar los nodos que pueden alcanzar más fácilmente al resto. Se puede interpretar como el tiempo de llegada a destino de algo que fluye a través de la red o como la rapidez que tomará la propagación de la información desde un nodo hasta los demás.

Nótese que cuanto mayor sea la distancia entre dos nodos, menor será la cercanía de estos. Dicha cercanía mide de cierta forma la accesibilidad de un nodo en la red. En una red valorada, la distancia está dada por los pesos de los arcos. Es por esto que no podemos tomar el número de viajes entre dos nodos como peso de los arcos, pues, en nuestro caso, un valor alto tiene una connotación positiva: cuántos más viajes hay entre un nodo y su destino, más *unidos* se encuentran. Por tanto, emplearemos el inverso de los pesos para valorar los arcos.

Se emplea la función `closeness_centrality()` y se especifica el argumento `distance = "Weight-Inverse"` para indicar cómo valorar los arcos. Puede verse la distribución mediante un histograma (Fig 1.15.)

Por la cola del ranking de closeness, tenemos estaciones ya conocidas. Las 2 estaciones con menor closeness se corresponden con:

- Estación 183 con 10.411126 . Situada en la Calle de Hermosilla, en el barrio Fuente del Berro, se enmarca en el distrito de Salamanca. Se corresponde también con la estación a la que menos viajes llegan y con la estación de la que menos viajes parten.
- Estación 201 con 12.129967. Situada en la calle José de Abascal,2, en la parte del barrio de Vallehermoso. También es de las estaciones con menor weighted in-degree y con menor weighted out-degree.

Los 2 nodos con más closeness se corresponden con:

- Estación 163 con 37.798761. Situada en Paseo de la Esperazna, en el barrio de Delicias. Se corresponde también con la estación con más in-degree weighted y out-degree weighted.
- Estación 149 con 37.775352. Situada en el Paseo de La Habana en el barrio de El Viso, se encuentra cerca del estadio Santiago Bernabéu y la zona de Nuevos Ministerios. Ocupa la posición número 13 en cuanto a estaciones con mayor weighted in-degree y la misma posición 13 en cuanto a estaciones con mayor weighted out-degree.

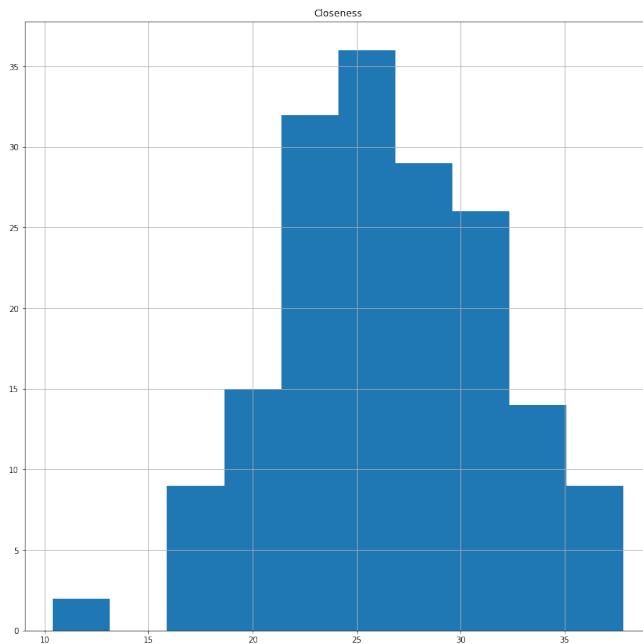


Figura 1.15: Histograma para el closeness.

Cabe destacar también la posición de algunas estaciones que ya hemos tratado como la número 129 de la Calle de José Abascal, que ocupa la cuarta posición en el ranking de closeness y se alzaba como la segunda con mayor in-degree weighted y out-degree weighted. Llama la atención la diferencia con la estación 201 a pesar de su proximidad geográfica. ¿Por qué una la estación 129 es tan usada y la estación 201 tan poco usada a pesar de estar al lado?

En Fig 1.16a, cuanto mayor es el tamaño y más oscuro es el nodo, mayor es el closeness. En Fig 1.16b se han representado en rojo las estaciones con closeness por debajo del percentil 10 y, en verde, aquellas con closeness por encima del percentil 90. No se observa ningún patrón espacial más allá de que, nuevamente, las estaciones de barrios de gran poder adquisitivo como Salamanca y Retiro poseen poco closeness.

En un grafo desconexo, el closeness todos los nodos será nulo ya que siempre existiría algún otro nodo para el cual la distancia con él resulte infinita -al no estar unidos, se asigna la mayor distancia posible-. Por tanto, esta medida tiene la desventaja de que sólo se puede aplicar sobre grafos conexos o componentes conexas en caso de redes no dirigidas y sobre componentes fuertemente conexos para redes dirigida. Recuérdese, la red de estudio es en sí fuertemente conexa.

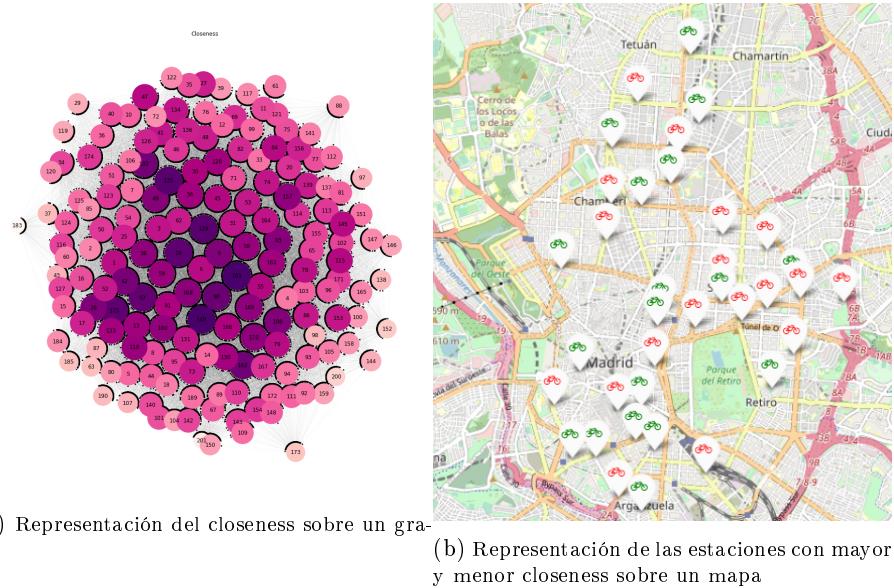


Figura 1.16: Visualizaciones para el closeness.

### Betwenesss.

Sea el grafo  $G = (V, E)$  donde  $|V| = N$ . El betweenness de un nodo  $u$  se define como:

$$C_B(u) = \sum_{j \neq k \in V} \frac{b_{juk}}{b_{jk}} \quad (3.2)$$

donde  $b_{juk}$  es el número de caminos más cortos desde el nodo  $j$  hasta que nodo  $k$  que pasan por  $u$ ;  $b_{jk}$  es el número de caminos más cortos desde  $j$  hasta  $k$ .

Se trata de una medida de centralidad que cuantifica la frecuencia con la que un nodo se encuentra dentro de los caminos más cortos entre otros nodos. A los caminos más cortos se les conoce como geodésicas. La idea intuitiva es que si se eligen dos nodos al azar y posteriormente uno de los caminos más cortos entre ellos, entonces los nodos con mayor betweenness serán aquellos que aparezcan con mayor probabilidad dentro del camino escogido.

Puesto que nuevamente se involucran distancias, hemos de escoger la columna "Weights-Inverse" como los pesos de los arcos, algo que especificamos en la función `betweenness_centrality()` mediante el argumento `weights = "Weight-Inverse"`. En grafos dirigidos, networkx normaliza por  $1/((N - 1)(N - 2))$ . Puede verse la distribución mediante un histograma (Fig 1.17).

Como puede intuirse a partir del histograma, son muchas las estaciones que poseen un betweenness nulo, en concreto 61. Entre ellas, hemos de volver a citar, por ejemplo, a la estación 201. Por el contrario, estaciones como la 163 o 129 vuelven a situarse como las estaciones con más betweenness. Destaca la estación 175 con un betweenness de 0.115946, que se sitúa en la Calle de Jaime El Conquistador, en el barrio de Delicias.

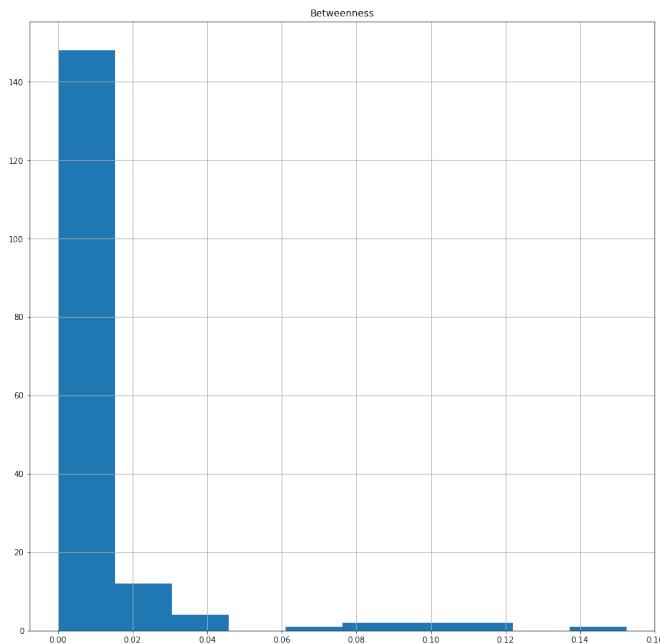


Figura 1.17: Histograma para el betweenness.

En Fig 1.18a, cuanto mayor es el tamaño y más oscuro es el nodo, mayor es el betweenness. En Fig 1.18b se han representado en rojo las estaciones con betweenness por debajo del percentil 10 y, en verde, aquellas con betweenness por encima del percentil 90. No se observa ningún patrón espacial más allá de que, nuevamente, las estaciones de barrios de gran poder adquisitivo como Salamanca y Retiro poseen poco betweenness. Son muy abundantes las estaciones con betweenness nulo y se distribuyen espacialmente por toda la ciudad.

### Eigenvector centrality.

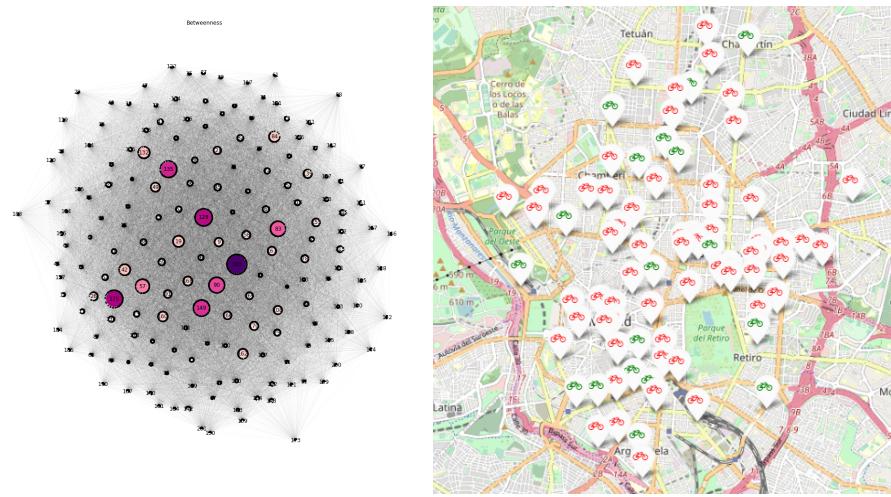
Sea el grafo  $G = (V, E)$ . El eigenvector centrality calcula la centralidad de un nodo en base a la centralidad de sus nodos vecinos. El eigenvector centrality de un nodo  $u$  se define como:

$$Ax = \lambda x$$

donde  $A$  es la matriz de adyacencia del grafo y  $\lambda$  es autovalor. Por el teorema de Perron-Frobenius, existe una solución única y positiva si  $\lambda$  es el mayor autovalor asociado al autovector de la matriz de adyacencia  $A$ .

Es fundamental entender que es la matriz de adyacencia de  $G$ . Se trata de una representación matricial en el que la entrada  $a_{ij}$  de la matriz de adyacencia  $A$  se corresponde con el peso del arco que une el nodo  $i$  con el nodo  $j$ .

En esta ocasión sí nos interesa el número total de viajes entre las estaciones para llenar la matriz de adyacencia, lo que indicamos con el argumento `weight="Weight"` de la función `eigenvector_centrality()`.



(a) Representación del betweenness sobre un grafo.  
 (b) Representación de las estaciones con mayor y menor betweenness sobre un mapa

Figura 1.18: Visualizaciones para el betweenness.

Intuitivamente, los nodos que poseen un alto valor de esta medida de centralidad están conectados a otros nodos que a su vez son muy relevantes en el sentido de la misma medida o bien a muchos otros nodos quizás menos relevantes. Por el contrario, los nodos conectados a otros nodos periféricos o poco relevantes tendrán un valor bajo de esta medida. Luego los nodos con alto valor para esta medida serían, en otro contexto, buenos candidatos para divulgar información. En el caso de la red de BiciMAD, se tratan de estaciones conectadas a su vez con otras relevantes.

Las estaciones 135, 163 y 129 vuelven a copar los valores más altos para esta medida de centralidad del mismo modo que las estaciones 183, 73, 39 y 152 vuelven a ocupar la parte baja del ranking.

### PageRank.

La mayoría de las variaciones del eigenvector centrality buscan establecer ajustes de parámetros para encontrar soluciones finitas del sistema de ecuaciones de la definición original de la medida. Hay algunas variaciones adicionales que se inspiran en la medida original para aplicaciones en dominios específicos, como es el caso de PageRank, creado originalmente para mejorar el motor de búsqueda de Google.

Volvemos a emplear el número de viajes entre nodos como peso de los arcos, pues, la red de BiciMAD sigue el mismo principio por el que el motor de búsqueda de Google: una estación será más importante si a ella llegan mas viajes. También empleamos "Weights" en el cálculo del eigenvector centrality. Emplearemos la función `pagerank()` detallando el argumento `weight = "Weights"`. En la figura (Fig 1.20a) se observa la distribución de esta medida y en la imagen (Fig 1.20b)

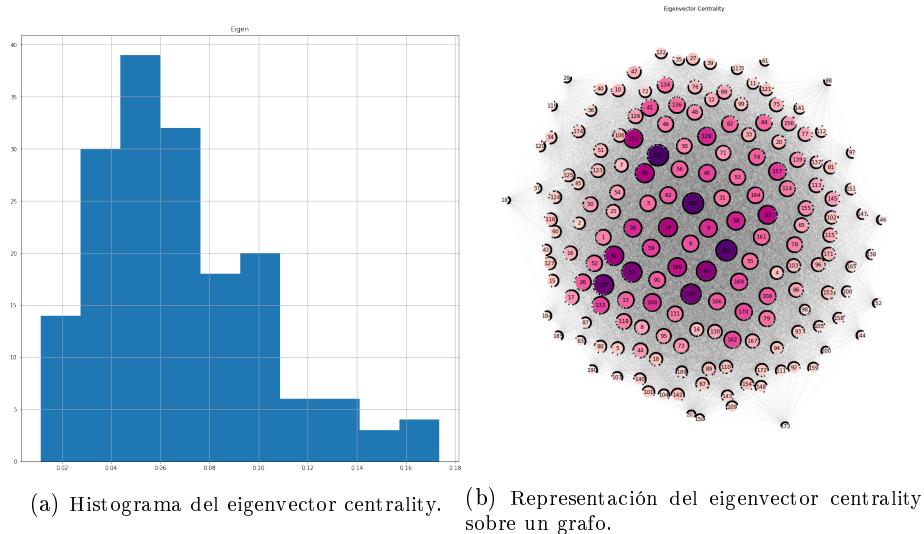


Figura 1.19: Visualizaciones para eigenvector centrality.

se dibuja sobre un grafo. Nuevamente, cuanto mayor y más oscuro es el nodo, mayor es el valor de esta medida de centralidad.

Nuevamente, las estaciones 163, 129, 135 y 90 son las que poseen un valor más alto de esta medida y, por la parte baja, reaparecen las estaciones 183, 29, 152, 173, 200.

## HITS

El algoritmo HITS calcula el authority de un nodo -basado en los arcos que llegan a él- y el hub -basado en los nodos que salen de él-.

La idea detrás de Hubs y Authorities surgió de una visión particular de la creación de páginas web. Ciertas páginas web conocidas como hubs sirven como grandes directorios que no son realmente autoritativos en la información que poseen sino que se usan como compilaciones de un amplio catálogo de información que redirige al usuario a otras páginas de más autoridad. Un hub representa una página que señala a otras muchas y un authority representa una web muy referenciada. Es por ello que debemos usar "Weights".

Podemos ver la distribución en sendos histogramas para los hubs (Fig 1.21a) y para los authorities (Fig 1.21b).

## Coeficiente de clustering.

El coeficiente de clustering de un nodo cuantifica cómo de agrupado o interconectado está con sus vecinos. Ya se ha comprobado que la red considerada es muy conexa ya que la mayoría de los nodos se conectan con el resto, por ello, esta medida quizás no sea la más descriptiva para esta red.

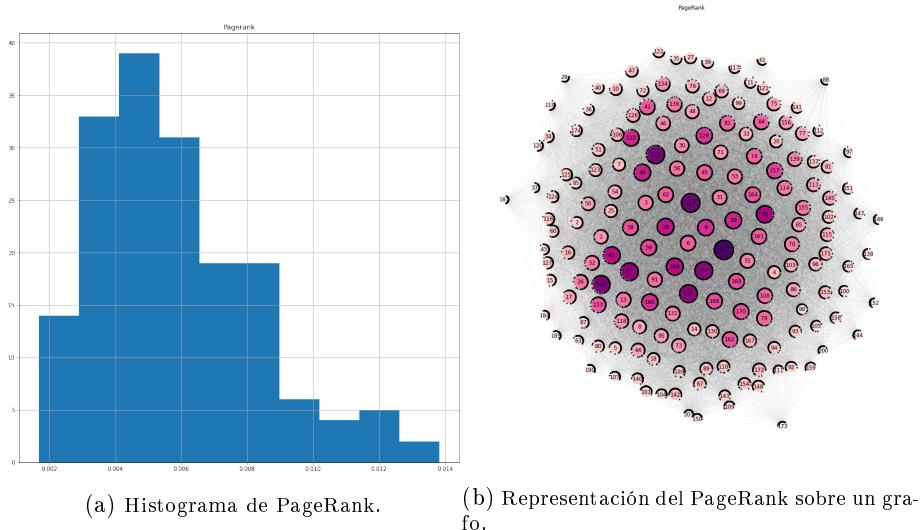


Figura 1.20: Visualizaciones para PageRank.

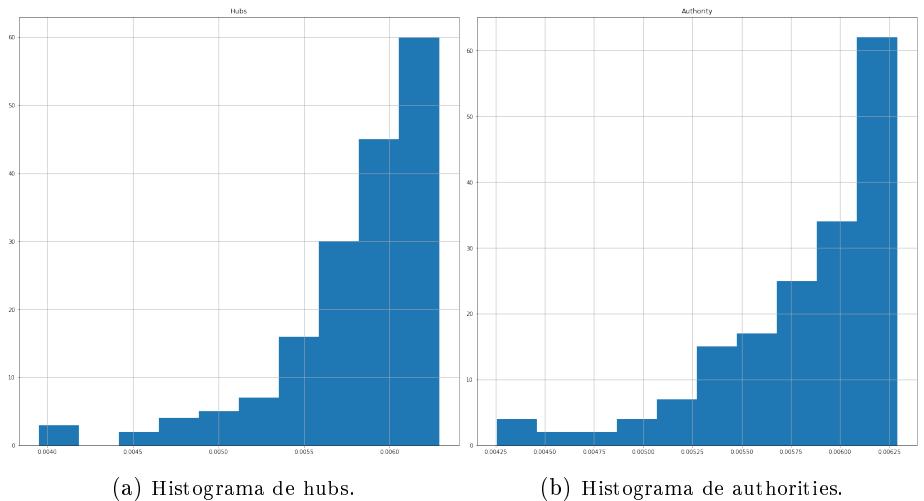


Figura 1.21: Visualizaciones para Hits.

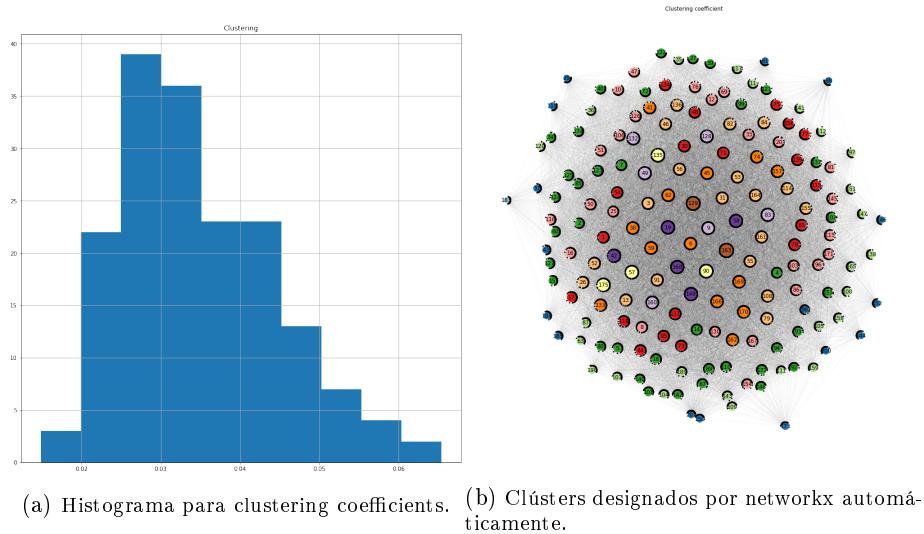


Figura 1.22: Visualizaciones para los coeficientes de clustering.

Para redes valoradas, existen diversas formas de definir el coeficiente de clustering. La librería *networkx* define el coeficiente de clustering de un nodo  $u$  como:

$$c_u = \frac{1}{deg(u)(deg(u) - 1)} \sum_{v,w \in V} (\hat{w}_{uv}\hat{w}_{uw}\hat{w}_{vw})^{1/3}$$

donde  $V$  es el conjunto de vértices del grado,  $deg(u)$  es el grado del nodo  $u$ ,  $\hat{w}_{ij} = w_{ij}/max(w)$  donde  $w_{ij}$  es el valor del arco que parte del nodo  $i$  hasta el nodo  $j$  y  $max(w)$  el máximo de los pesos. Nuevamente, usamos "Weight".

El coeficiente de clustering medio es de 0.034807888186262376 y a la vista del histograma de su distribución (Fig 1.22a) puede comprobarse que se halla entre 0.02 y 0.06. La variación es mínima. Por las alturas de los bins, podríamos considerar 8 ó 9 clusters, aunque no sería la medida más descriptiva para esta red. Se ha dejado que networkx dibuje un grafo distinguiendo clusters por color de manera automática (Fig 1.22b).

Nuevamente, las estaciones con un mayor coeficiente de clustering son las 163, 129, 175, 90 y las que tienen un coeficiente de clustering menor son 183, 29, 201, 200, 152.

Finalmente, todas estas medidas han sido agrupadas en un único dataframe y almacenadas en el archivo my\_results.csv



# Capítulo 2

## Neo4J.

Neo4j se trata de una base de datos NoSQL orientada a grafos que se compone de dos elementos fundamentales:

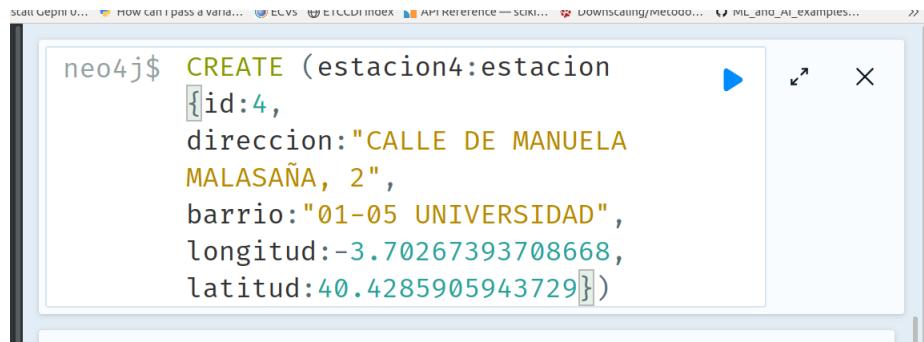
- **Nodos.** Representan entidades con un concepto de entidad único. En nuestro caso, se corresponden con las estaciones de BiciMAD.
- **Relaciones.** Representan conexiones o interacciones entre los nodos. En nuestro caso, se corresponden con los arcos que denotan el número de viajes entre dos nodos.

### 1. Construcción de la red social.

Tanto los nodos como las relaciones que constituyen los elementos de Neo4j pueden contener propiedades, que son equivalentes a las tablas de las columnas en el modelo relaciones. Recuérdese la estructura de nuestra red social de BiciMAD (Fig 1.3).

Se observa que tanto las estaciones de origen como destino -nodos- poseen 4 propiedades -latitud, longitud, dirección y barrio- además de un identificador propio. Por tanto, para definir una estación en la base de datos de Neo4j, habría que realizar una query como la mostrada en Fig 2.1. Mediante la operación *CREATE*, inicializamos la variable temporal *estacion4* en la que se almacena un nodo cuya etiqueta es **estacion**. Además, entre llaves, definimos las propiedades *barrio*, *direccion*, *latitud*, *longitud*, *id*.

Para enlazar dos estaciones, han de definirse relaciones. En este caso, serían arcos valorados con el número de viajes entre el nodo origen y el nodo destino. Ha de tener en cuenta que en Neo4j las relaciones son unidireccionales, algo que no supone un problema ya que la red de BiciMAD es dirigida. La operación *CREATE* también permite crear relaciones entre nodos. Se crea la relación *VIAJES\_HACIA* que contiene una propiedad *total* que almacena el número de viajes.



```
neo4j$ CREATE (estacion4:estacion
  {id:4,
  direccion:"CALLE DE MANUELA
  MALASAÑA, 2",
  barrio:"01-05 UNIVERSIDAD",
  longitud:-3.70267393708668,
  latitud:40.4285905943729})
```

Figura 2.1: Ejemplo de operación **CREATE** para inicializar un nodo estación.



```
1 CREATE (estacion4:estacion {id:4, direccion:"CALLE DE MANUELA MALASAÑA, 2",
  barrio:"01-05 UNIVERSIDAD", longitud:-3.70267393708668,
  latitud:40.4285905943729})
CREATE (estacion12:estacion {id:12, direccion:"CALLE DE SAN BERNARDO, 87",
  barrio:"01-05 UNIVERSIDAD", latitud:-3.70622023810713,
  longitud:40.4285272736706})
CREATE (estacion4)-[:VIAJES_HACIA {total:13}]->(estacion12)
```

Figura 2.2: Ejemplo de operación **CREATE** para inicializar una relación.

Las diferentes operaciones **CREATE** han de ejecutarse juntas para poder acceder a las variables temporales que apuntan a los nodos creados. En Fig 2.2, se crean los nodos *estacion4* y *estacion12* que son unidos mediante la relación *VIAJES\_HACIA* indicado que se han producido 13 viajes desde *estacion4* a *estacion12*. En resultado puede visualizarse en Fig 2.3.

Para establecer más variedad de nodos y relaciones, se crearán nodos etiquetados como *centroAdministrativo* que simularán ser oficinas de BiciMAD (Fig 2.4) y serán encargados de gestionar ciertas estaciones. Los nodos de tipo tendrán dos propiedades *centroAdministrativo*: un identificador y una dirección. La relación se modelará como *A\_CARGO\_DE* (Fig 2.5).

También se crearán unos nodos etiquetados como *persona* que simularán ser trabajadores de los nodos etiquetados como *centroAdministrativo* (Fig 2.7)

## 2. Consultas propuestas.

Si se juntan todos los nodos y relaciones descritos anteriormente, podemos dar lugar a una red como la mostrada en Fig 2.8. Se realizan las consultas propuestas sobre esta red social

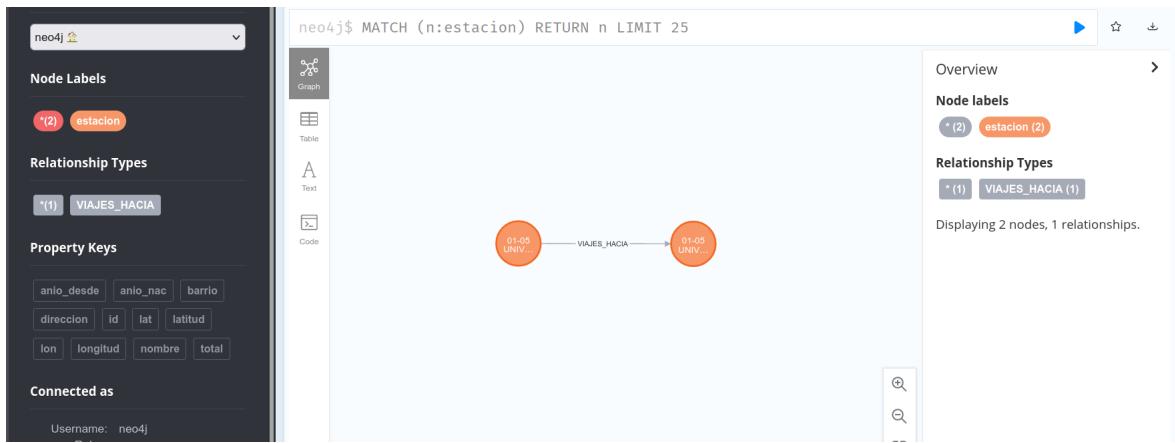


Figura 2.3: Resultado de ejecutar la query mostrada en Fig 2.2.



Figura 2.4: Ejemplo de creación de nodo etiquetado como centroadministrativo.

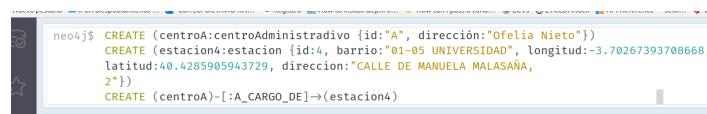


Figura 2.5: Ejemplo de creación de relación A\_CARGO\_DE.

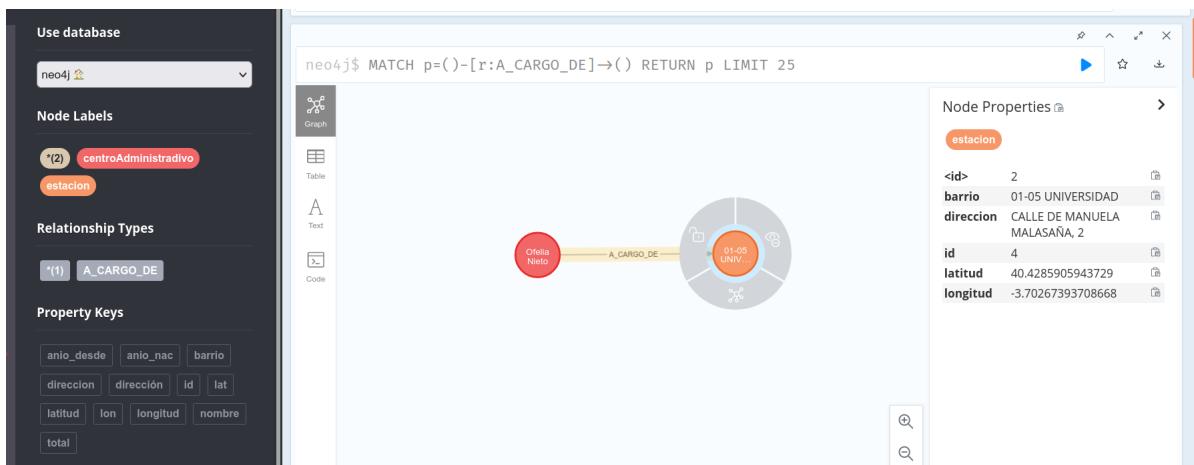


Figura 2.6: Resultado de ejecutar la query mostrada en Fig 2.5.

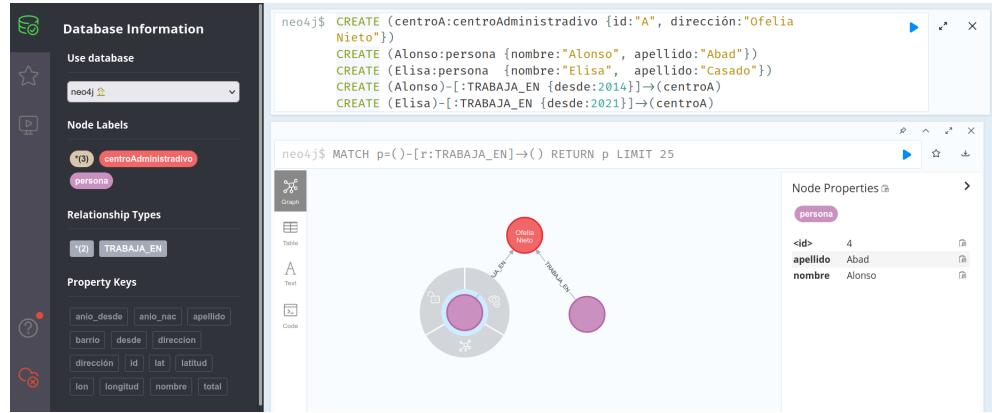


Figura 2.7: Creación de nodos etiquetados como persona relacionados con nodos centroAdministrativo mediante TRABAJA\_EN.

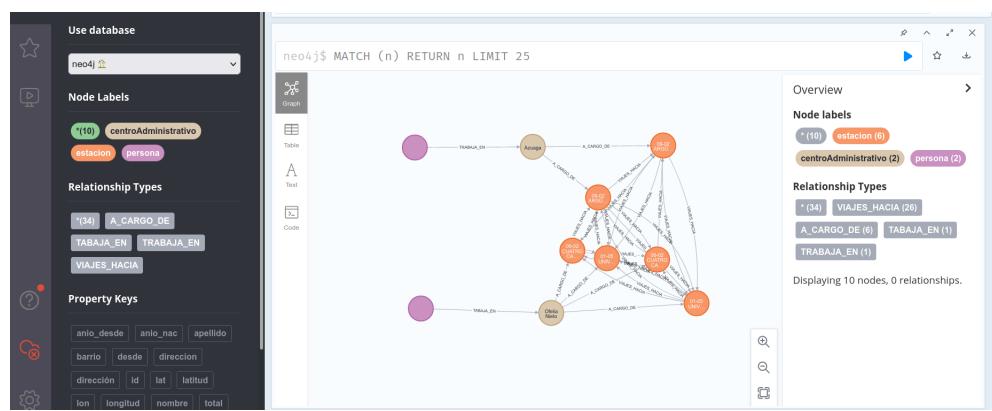


Figura 2.8: Red social en Neo4j.

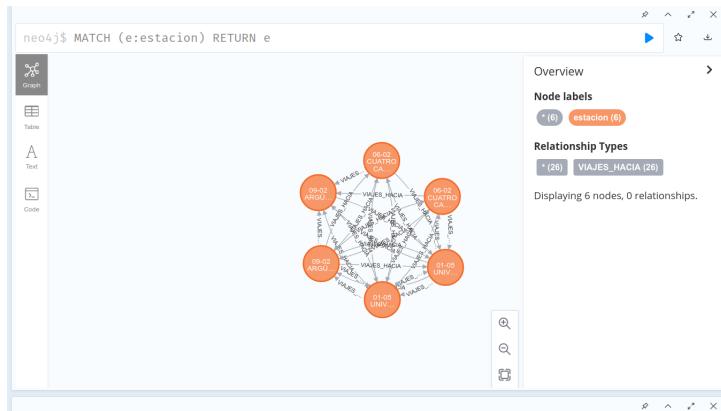


Figura 2.9: Consulta para retornar todos los nodos etiquetados como *estacion*.

### Retornar los nodos con una etiqueta concreta.

El operador *MATCH* permite realizar consultas con condiciones sobre los datos de los nodos y sus relaciones. La consulta de Fig 2.9 utiliza este operador para devolver todos los nodos *e*-variable en la que se almacenarán los nodos a retornar- etiquetados como *estacion*.

La consulta para devolver los nodos etiquetados *centroAdministrativo* o *persona* sería análoga. Sólo habría que sustituir *estacion* por el tipo de nodo que se quiera devolver.

### Retornar las relaciones de un tipo concreto.

Mediante la operación *MATCH*, también pueden realizarse consultas sobre relaciones. En Fig 2.10, se devuelven todas las relaciones etiquetadas como *A\_CARGO\_DE*. Devuelve la relación en sí y a los nodos involucrados en ella, en nuestro caso, los 2 centros administrativos como nodos origen y las 6 estaciones como destino. Pero también devuelve el resto de relaciones que involucran a los nodos devueltos, como son las relaciones *VIAJES\_HACIA* entre los nodos estaciones.

### Retornar todos los nodos que tengan una relación concreta con otro nodo.

En la consulta de Fig 2.11, se devuelven todos los nodos persona unidos mediante la relación *TABAJA\_EN* a un centro administrativo. Al crear la red cometí una falta de ortografía y hay dos relaciones que indican lo mismo: *TABAJA\_EN* y *TRABAJA\_EN*. Por eso sólo se nos devuelve una persona a pesar de que hay 2 que trabajan en centros administrativos.

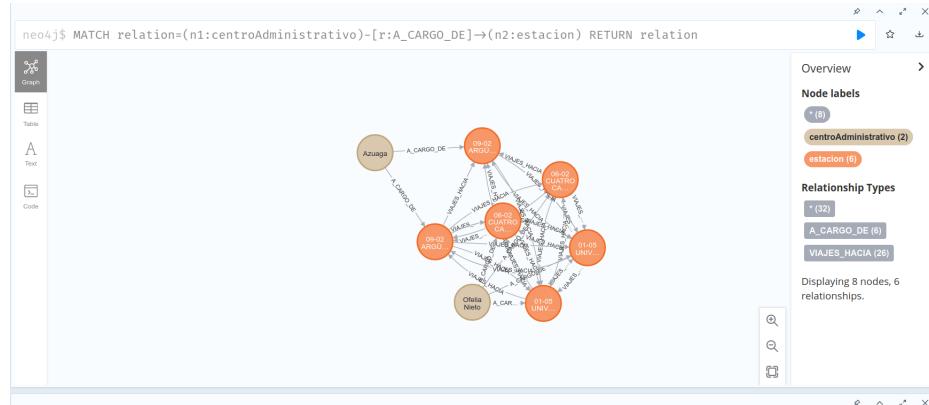


Figura 2.10: Consulta para devolver todas las relaciones de tipo A\_CARGO\_DE.



Figura 2.11: Consulta para devolver todas las relaciones de tipo A\_CARGO\_DE.

```
neo4j$ MATCH(p:estacion) RETURN count(*)
+-----+
| count(*) |
+-----+
| 6        |
+-----+
Started streaming 1 records after 1 ms and completed after 1 ms.
```

Figura 2.12: Consulta para contar todos los nodos etiquetados como estacion.

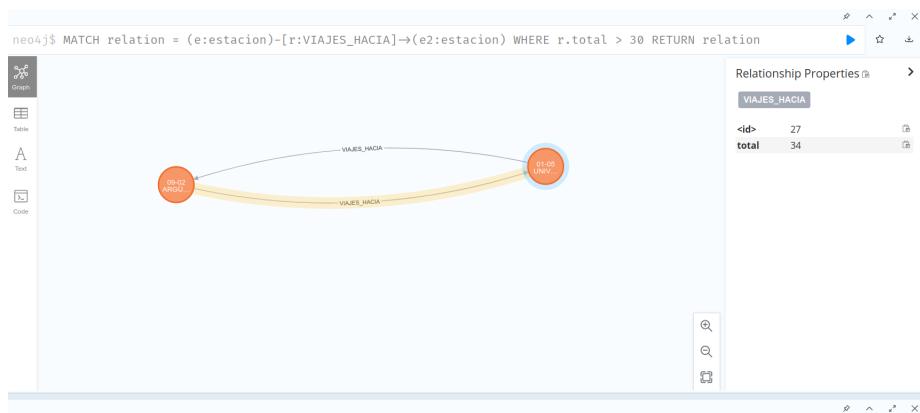


Figura 2.13: Consulta para contar todos los nodos etiquetados como estacion.

## Contar el número de nodos de un determinado tipo.

Para contar todos los nodos etiquetados como *estacion* realizamos la consulta mostrada en Fig 2.12

## Otras consultas.

Al igual que el SELECT de SQL, la operación *MATCH* en Cypher tiene la cláusula *WHERE* para expresar condiciones. La consulta de Fig 2.13 impone una condición sobre la relación *VIAJES\_HACIA*. Para ello, accede a la propiedad *total* de la relación e impone que se devuelvan aquellas con más de 30 viajes. Como puede verse, devuelve la relación, los nodos involucrados en ella y, además, el resto de relaciones que involucran a los nodos -como nos ocurría anteriormente-.

También pueden imponerse condiciones sobre los nodos. Por ejemplo, en la consulta de Fig 2.14, se devuelven todos los nodos etiquetados como *persona* que se llaman Alonso.

The screenshot shows the Neo4j browser interface. On the left, there is a sidebar with icons for Graph, Table, Text, and Code. The main area displays a single node highlighted in purple. To the right, the 'Node Properties' panel is open, showing the following data for the node:

persona
<id> 9
apellido Abad
nombre Alonso

At the bottom of the browser window, there are standard navigation buttons: back, forward, search, and close.

```
neo4j$ MATCH (p:persona) WHERE p.nombre = "Alonso" RETURN p
```

Figura 2.14: Consulta para contar todos los nodos etiquetados como estacion.

## Capítulo 3

# Conclusiones y trabajo futuro.

Los sistemas de bike-sharing permiten que los usuarios alquilen una bicicleta de manera temporal situada en cierta estación y, finalizado su uso, devolverla en otra estación del propio sistema. Se basan en el siguiente principio: los usuarios emplean las bicicletas según sus circunstancias propias sin tener que hacerse cargo de los gastos y responsabilidades que conlleva tener una bicicleta propia.

El aumento de estos servicios y el fomento de los datos en abierto permiten estudiar las dinámicas de la red y los motivos por los que los usuarios deciden hacer uso de las bicicletas así como permite ofrecer criterios para la toma de decisiones. Este análisis permite hacer frente a dos tipos de problemas principales:

- Problemas relacionados con el diseño e infraestructura del sistema bike-sharing. La planificación sobre dónde situar las estaciones y cuántas bicicletas poner a disposición del usuario debe tener en cuenta desde factores topográficos de la ciudad hasta la previsión de uso o principios de justicia social. No sólo deben tenerse en cuenta en la creación de la red sino durante su uso para mejorar la infraestructura. Es de especial interés el estudio del tipo de usuario -¿en qué edad se mueven la mayoría de ellos?- y el de patrones temporales -¿en qué momentos del día se requieren más bicicletas?, ¿quizá a la entrada del trabajo?-. En el **estudio previo** se abordó ligeramente este problema. Puede visualizarse un gráfico lineal (Fig 3.1) según la edad del usuario y un heatmap (Fig ??) que muestra el número de viajes totales producidos durante el mes de junio de 2019 distinguiendo por día de la semana y tramo horario.
- Problemas relacionados con el fomento de la bicicleta en la ciudad. Modificar la estructura de la propia ciudad y poner a disposición del ciudadano carriles bici o similares puede ser de gran ayuda. Incentivar el uso de los sistemas bike-sharing puede, quizás, depender de una política de precios flexible en la que estos varíen en función de la demanda, la climatología o la celebración de eventos en la ciudad. Para afrontar estos problemas, sería de interés cruzar la información de la red con datos de índole meteórológico, económico, o relacionados con el tráfico.

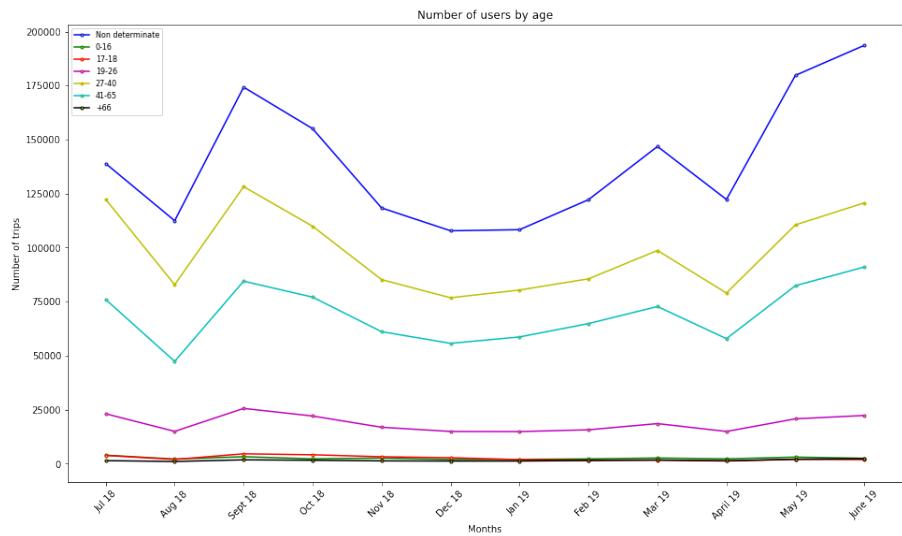


Figura 3.1: Gráfico de líneas distinguiendo según edad del usuario en el periodo comprendido entre julio de 2018 y junio de 2019.

Hemos limitado el Análisis de Redes Sociales únicamente a junio de 2019, de modo que también sería de interés ampliar el periodo temporal. Quizá con ello puedan establecerse criterios más adecuados a la hora de hacer el clustering.

Como se ha citado anteriormente, expandir el periodo temporal permitiría conocer mejor el tipo de usuario, los momentos del día en los que los viajes son más frecuentes y muchas otras cuestiones de interés.

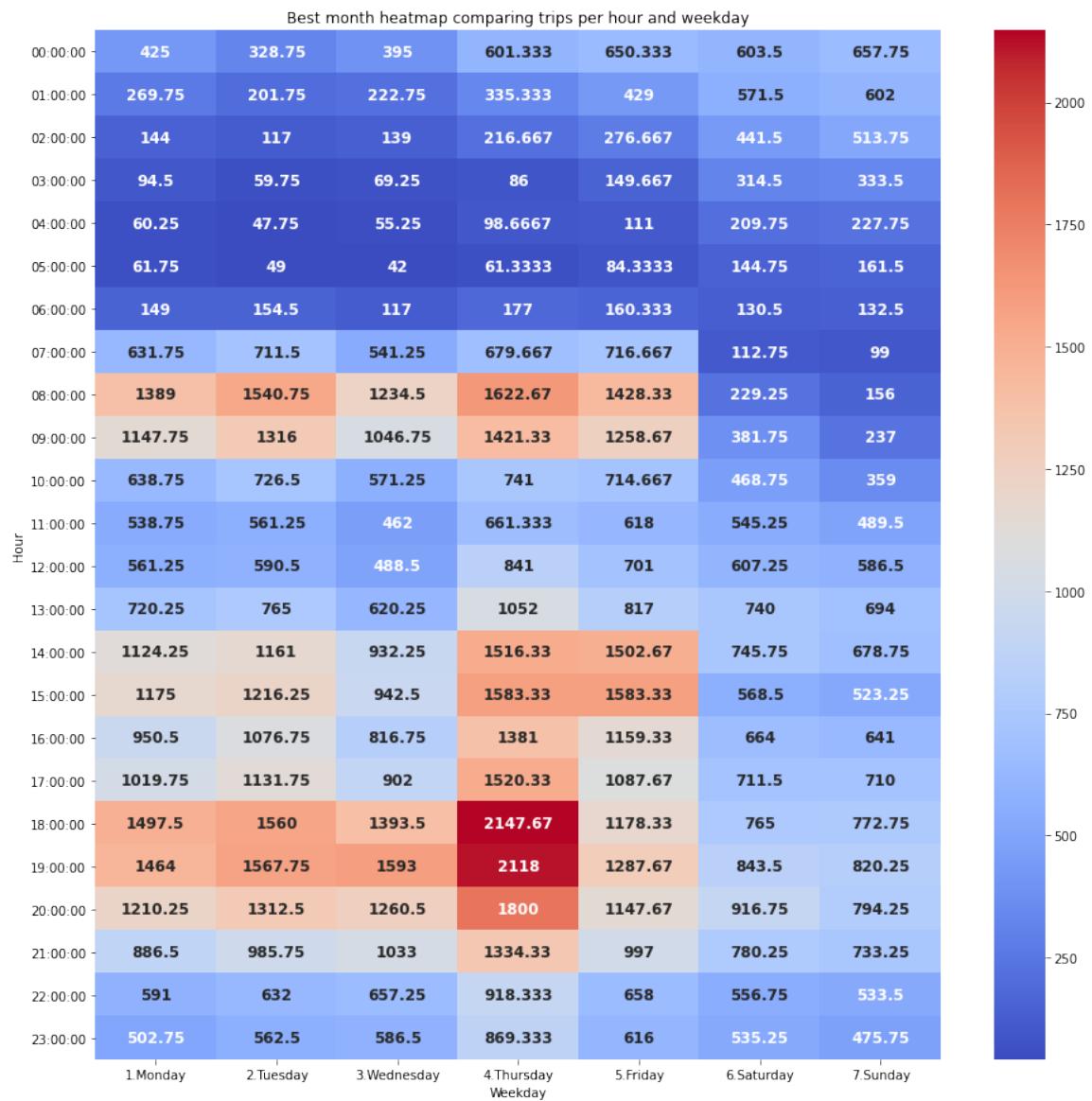


Figura 3.2: Heatmap para junio de 2019 mostrando el total de viajes producidos según día de la semana y tramo horario.



# Anexo

Todos los archivos empleados y generados durante la realización de este trabajo pueden consultarse en el siguiente [repositorio de GitHub](#).

El archivo bicimad.pdf se corresponde con la memoria del trabajo.

El archivo bicimad.gephi contiene los resultados del análisis de redes sociales parcialmente.

El archivo bicimad.ipynb contiene los resultados completos del análisis de redes sociales.

La carpeta datasets.zip contiene 3 datasets necesarios para la creación de la red:

- Bicimad\_Stations\_201906.json -> datos ofrecidos por BiciMAD para junio de 2019.
- bases\_bicimad.json -> datos relativos a la ubicación geográfica de las bases y estaciones.
- bases\_bicimad.xlsx -> datos relativos a la ubicación geográfica de las bases y estaciones.

La carpeta resultados.zip contiene 2 datasets:

- my\_graph.csv -> datos a partir de los cuales generar la red dirigida y valorada.
- my\_results.csv -> valores de las medidas de centralidad consideradas.



# Bibliografía

- [DD16] Ewa Dobrzyńska y Maciej Dobrzyński. «Structure and dynamics of a public bike-sharing system. Case study of the public transport system in Białystok». En: *Ekonomia i Zarządzanie* 8.4 (2016).
- [Gar22] Diego García Sáiz. *Apuntes de DataLab Ciencias Sociales*. 2022.
- [GPZ14] Diego Garcia-Saiz, Camilo Palazuelos y Marta Zorrilla. «Data mining and social network analysis in the educational field: An application for non-expert users». En: *Educational data mining*. Springer, 2014, págs. 411-439.
- [OAS10] Tore Opsahl, Filip Agneessens y John Skvoretz. «Node centrality in weighted networks: Generalizing degree and shortest paths». En: *Social networks* 32.3 (2010), págs. 245-251.
- [Tle+20] Aizhan Tlebaldinova y col. «Temporal Network Approach to Explore Bike Sharing Usage Patterns.» En: *VEHITS*. 2020, págs. 129-136.