

Proyecto Fin de Carrera

Ingeniería de las Tecnologías Industriales

Análisis de la vulnerabilidad de la red viaria urbana de Sevilla utilizando teoría de redes complejas.

Autor: Álvaro Díaz Lemus

Tutor: José David Canca Ortiz

Dpto. de Organización y Gestión de Empresas I
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2023



Proyecto Fin de Carrera
Ingeniería de Tecnologías Industriales

**Análisis de la vulnerabilidad de la red viaria
urbana de Sevilla utilizando teoría de redes
complejas.**

Autor:
Álvaro Díaz Lemus

Tutor:
José David Canca Ortiz
Catedrático de Universidad

Dpto. de Organización Industrial y Gestión de Empresas I
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2023

Proyecto Fin de Carrera: Análisis de la vulnerabilidad de la red viaria urbana de Sevilla
utilizando teoría de redes complejas.

Autor: Álvaro Díaz Lemus
Tutor: José David Canca Ortiz

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2023

A mi familia
A mis amigos
A mi país

Agradecimientos

A mi familia por el apoyo desde el minuto uno en la escuela y por ser el motivo de mi esfuerzo diario. A mis amigos por la confianza en mí y, por haberme guiado hasta conseguir terminar el grado. Gracias a José David Canca por darme la oportunidad y la ayuda necesaria para la realización del trabajo.

*Álvaro Díaz Lemus
Sevilla, 2023*

Resumen

Las redes de transporte son fundamentales no solo en el ámbito económico y en la infraestructura pública, sino también por que son la vía de rescate y de socorro ante situaciones de emergencia. En el presente trabajo nos centramos en la red viaria de transporte de la ciudad de Sevilla, ciudad con una infraestructura y un casco antiguo de los mas antiguos de las ciudades de España. No obstante, las medidas y procedimientos son de carácter general y el análisis propuesto podría ser aplicado a otra ciudad cualquiera.

El problema que nos interesa es conocer en que parte, nodos y arcos de la red, hay que prestar mas atención de cara a evitar incidencias, al tratarse de los componentes más vulnerables de la red ante situaciones adversas y, por tanto, en caso de bloqueo podrían llegar a provocar un colapso en el movimiento de los ciudadanos. Para el desarrollo del proyecto se ha usado la herramienta gráfica yEd, para visualizar y manejar el grafo de la red de Sevilla, la librería Networkx de Python para calcular las medidas debido a la gran cantidad de datos que es necesario manejar y Excel como soporte de almacenamiento de los datos que se usarán en el código en lenguaje Python.

Abstract

Transport networks are essential not only in the economic and public infrastructure realms but also because they serve as a means of rescue and relief in emergency situations. In this study, we focus on the road transportation network of the city of Seville, a city with one of the oldest infrastructures and historic districts in Spain. However, the measures and procedures discussed are of a general nature, and the proposed analysis could be applied to any other city.

The problem of interest is understanding which parts, nodes, and arcs of the network require more attention to prevent incidents, as they are the most vulnerable components of the network in adverse situations. Therefore, a blockage in these components could potentially lead to a collapse in the movement of citizens. For the project's development, the graphical tool yEd was used to visualize and manage the graph of the Seville network, the Python Networkx library to calculate measures due to the large amount of data that needs handling, and Excel as a support for storing the data used in the Python code.

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xvii
Índice de Figuras	xix
1 Introducción	1
2 Teoría de Grafos	5
2.1 <i>Terminología y conceptos básicos</i>	5
2.2 <i>Caracterización de redes complejas</i>	5
2.2.1 <i>Métricas básicas</i>	6
2.3 <i>Medidas de centralidad</i>	10
3 Revisión de la Literatura	15
3.1 <i>Concepto de fiabilidad</i>	15
3.2 <i>Medidas en las que basar la fiabilidad de una red de transporte</i>	15
3.3 <i>Vulnerabilidad de las redes de transporte</i>	16
3.4 <i>Vulnerabilidad de redes por carreteras urbanas</i>	16
3.5 <i>Redes de Transporte público</i>	18
3.6 <i>Relación entre la topología de la red y el flujo del tráfico</i>	19
4 Medidas en redes con Pesos	21
4.1 <i>Medidas de centralidad en redes urbanas con datos</i>	21
5 Aplicación de red de vehículo privado de la ciudad de Sevilla	23
5.1 <i>Parque automovilístico</i>	23
5.2 <i>Movilidad urbana</i>	24
5.3 <i>Viario</i>	25
5.3.1 <i>Jerarquización del viario</i>	25
5.4 <i>Circulación</i>	27
5.4.1 <i>Análisis del tráfico interior</i>	27
6 Implementación y resultados	29
6.1 <i>Recopilación de datos</i>	29
6.2 <i>Librería Networkx</i>	32
6.3 <i>Resultados</i>	33
6.4 <i>Rankings</i>	38
6.4.1 <i>Comparación entre índices de Networkx</i>	38
6.4.2 <i>Número de caminos mínimos por arco</i>	44
6.4.3 <i>Flujo de personas que circulan por la red</i>	45
7 Conclusiones	49
Bibliografía	51
Anexo. Código de Python	54
<i>Programa 1. Resolución del cálculo de los índices.</i>	54
<i>Programa 2. Gráficos y Visualizaciones.</i>	64

ÍNDICE DE TABLAS

TABLA 1 MEDIDAS DE CENTRALIDAD POR CATEGORÍAS. FUENTE: (FREEMAN, 1977).	11
TABLA 2 VÍAS PRINCIPALES DE MAYOR FLUJO. FUENTE: (IMD, 2016)	28
TABLA 3 DESCRIPCIÓN DE LOS ARCOS QUE COMPONEN LA RED. FUENTE: ELABORACIÓN PROPIA.	30
TABLA 4 ARCOS FORMADOS POR LOS NODOS DE MÁS VALOR Y SU LOCALIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	38
TABLA 5 ARCOS FORMADOS POR LOS NODOS DE MAYOR VALOR Y SU LOCALIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	39
TABLA 6 ARCOS FORMADOS POR LOS NODOS DE MAYOR VALOR Y SU LOCALIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	40
TABLA 7 ARCOS FORMADO POR LOS NODOS DE MAYOR VALOR Y SU LOCALIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	41
TABLA 8 ARCOS FORMADOS POR LOS NODOS DE MAYOR VALOR Y SU LOCALIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	42
TABLA 9 ARCOS FORMADOS POR LOS NODOS DE MAYOR VALOR Y SU LOCALIZACIÓN. FUENTE: ELABORACIÓN PROPIA.	43
TABLA 10 RANKING DE LOS 50 ARCOS POR LOS CUALES PASAN MÁS CAMINOS MÍNIMOS. FUENTE: ELABORACIÓN PROPIA.	44
TABLA 11 RANKING DE LOS 50 ARCOS CON MAYOR FLUJO DE PERSONAS Y SU LOCALIZACIÓN. FUENTE: ELABORACIÓN PROPIA.....	45
TABLA 12 RANKING DE LOS 50 NODOS CON MAYOR FLUJO DE PERSONAS. FUENTE: ELABORACIÓN PROPIA.	47

ÍNDICE DE FIGURAS

ILUSTRACIÓN 1 GRÁFICO DE PORCENTAJES DE LOS VEHÍCULOS UTILIZADOS EN SEVILLA CAPITAL. FUENTE: (IMD, 2016).	2
ILUSTRACIÓN 2 FORMACIÓN DE TRIÁNGULOS EN RED NO DIRIGIDA. FUENTE: ELABORACIÓN PROPIA.	9
ILUSTRACIÓN 3 EJEMPLO DE NODO CON VALOR MÁS ALTO DE CLOSENNES CENTRALITY TOPOLÓGICO. FUENTE: (FREEMAN, 1977).	12
ILUSTRACIÓN 4 EJEMPLO DE NODO CON EL VALOR MÁS ALTO DE BETWEENNES CENTRALITY TOPOLÓGICO. FUENTE: (FREEMAN, 1977)...	12
ILUSTRACIÓN 5 EJEMPLO DE NODO CON EL VALOR MÁS ALTO DE EIGENVECTOR CENTRALITY TOPOLÓGICO. FUENTE: (FREEMAN, 1977).	13
ILUSTRACIÓN 6 MAPA DE ESTUDIO CON CARRILES COMO ATRIBUTOS. FUENTE: ELABORACIÓN PROPIA.	22
ILUSTRACIÓN 7 MAPA DE ESTUDIO CON VELOCIDADES COMO ATRIBUTOS. FUENTE: ELABORACIÓN PROPIA.	22
ILUSTRACIÓN 8 PORCENTAJES DE LOS DIFERENTES TIPOS DE VEHÍCULOS EN LA CIUDAD DE SEVILLA. FUENTE: (IMD, 2016).	23
ILUSTRACIÓN 9 EVOLUCION DEL PARQUE DE VEHÍCULOS DE SEVILLA. FUENTE: (IMD, 2016).	23
ILUSTRACIÓN 10 DISTRITOS DE SEVILLA. FUENTE: (IMD, 2016).	24
ILUSTRACIÓN 11 EJES Y RONDAS EXTERIORES. FUENTE: (IMD, 2016).	25
ILUSTRACIÓN 12 DISTRIBUCIÓN DEL VIARIO PRINCIPAL Y SECUNDARIO DE LA CIUDAD. FUENTE: (IMD, 2016).	26
ILUSTRACIÓN 13 ÍNDICE DE MOTORIZACIÓN. FUENTE: (IMD, 2016).	27
ILUSTRACIÓN 14 MAPA DE ESTUDIO VISTO EN LA APLICACION YEd. FUENTE: ELABORACIÓN PROPIA.	29
ILUSTRACIÓN 15 MAPA CON LOS NODOS CLASIFICADOS SEGÚN SU VALOR. FUENTE: ELABORACIÓN PROPIA.	33
ILUSTRACIÓN 16 MAPA CON LOS NODOS CLASIFICADOS SEGÚN SU VALOR. FUENTE: ELABORACIÓN PROPIA.	34
ILUSTRACIÓN 17 MAPA CON LOS NODOS CLASIFICADOS SEGÚN SU VALOR CONSIDERANDO ATRIBUTOS. FUENTE: ELABORACIÓN PROPIA.	34
ILUSTRACIÓN 18 MAPA CON LOS NODOS CLASIFICADOS SEGÚN SU VALOR CONSIDERANDO ATRIBUTOS. FUENTE: ELABORACIÓN PROPIA.	35
ILUSTRACIÓN 19 MAPA CON LOS NODOS CLASIFICADOS SEGÚN SU VALOR. FUENTE: ELABORACIÓN PROPIA.	36
ILUSTRACIÓN 20 NÚMERO DE CAMINOS MÍNIMOS QUE PASAN POR CADA ARCO. FUENTE: ELABORACIÓN PROPIA.	36
ILUSTRACIÓN 21 FLUJO DE PERSONAS QUE CIRCULAN POR CADA ARCO. FUENTE: ELABORACIÓN PROPIA.	37
ILUSTRACIÓN 22 FLUJO DE PERSONAS QUE ATRAVIESAN CADA NODO. FUENTE: ELABORACIÓN PROPIA.	37

1 INTRODUCCIÓN

Nuestras sociedades, como describe (Mattsson & Jenelius, 2015), dependen en gran medida de una serie de sistemas de infraestructura críticos, como la energía eléctrica, el transporte, el suministro de agua y el manejo de aguas residuales, la información y la comunicación, y los sistemas bancarios. Estos sistemas se han vuelto progresivamente cada vez más complejos e interdependientes. Por ejemplo, la mayoría de ellos requieren energía eléctrica, acceso a redes informáticas y conectividad por carretera. Si el suministro de cualquiera de estos servicios se detiene o se reduce drásticamente, los sistemas dependientes fallarán o funcionarán con un bajo nivel de rendimiento.

El transporte es importante en el sentido de que permite a las personas participar en sus propias actividades humanas. Con el crecimiento de la población, la demanda del transporte también está creciendo a su vez. Esto nos lleva a que haya más tráfico, lo que crea problemas de congestión en el flujo de transporte, además de contaminación acústica y del aire y más accidentes. Hay que planificar adecuadamente las redes de transporte urbanas para garantizar la movilidad sostenible, con el objetivo de facilitar la movilidad de la creciente población en la ciudad y en los extrarradios (Zanjirani Farahan & Miandoabchi, 2013).

En este documento se desarrollará un análisis de la vulnerabilidad de la red viaria de transporte de Sevilla, a partir de una serie de índices que permitan deducir que nodos y arcos del viario tienen mayor impacto en el flujo del tráfico. Estudiar y analizar la vulnerabilidad del viario de la ciudad ayudará a priorizar la planificación, el presupuesto y el mantenimiento de las vías y, también será útil para preparar planes de respuesta de emergencia. No existe una definición comúnmente establecida en el sistema de transporte para la vulnerabilidad, sin embargo, podemos tomar algunas de ellas como ejemplos:

- Vulnerabilidad como la susceptibilidad a incidentes que pueden resultar en reducciones considerables en la capacidad de servicio por carretera (Berdica, 2002).
- Susceptibilidad a riesgos “raros y grandes” (Gallagher, 2015).
- (Taylor, 2003) desde la perspectiva de la accesibilidad, cuando se daña un pequeño número de enlaces hacia un nodo de la red, la accesibilidad de ese nodo se reduce considerablemente, lo que indica que es punto es frágil.

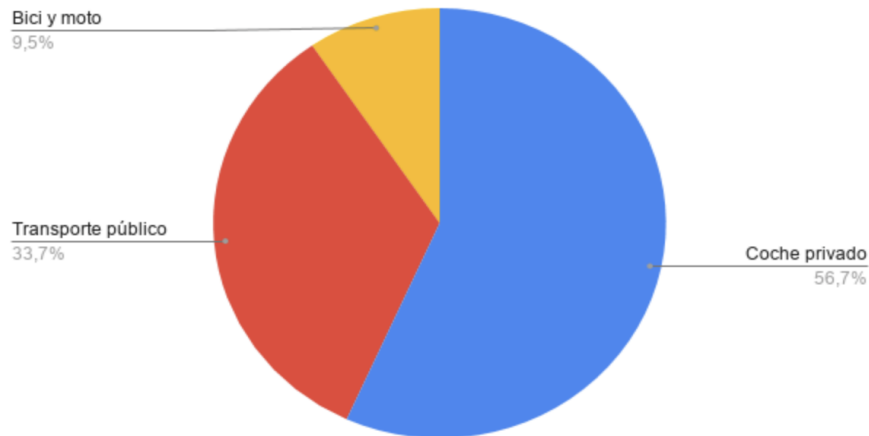
El uso de la teoría de redes complejas y la extracción de información a partir de datos basados en grafos ha aumentado su popularidad durante los últimos 10 años. Esto se debe a la versatilidad demostrada por estas herramientas en amplios campos como biología, economía y ciencias sociales (Newman, 2003). El enfoque en redes no solo es útil para simplificar y visualizar una gran cantidad de datos, sino que también es eficaz para encontrar elementos destacados dentro de la red.

Durante décadas se ha prestado mucha atención a la investigación de los sistemas de transporte debido a su importancia. En la mayoría de estos estudios el modelo de red se representa con

nodos como cruces o conexiones. Estos cruces se conectan mediante arcos que representan las rutas seguidas entre conexiones consecutivas.

En Sevilla circulan diariamente 399.284 vehículos de los cuales el 53,7% transitan en el interior de la propia capital, nuestra zona de estudio, ver ilustración 1.

Ilustración 1 Gráfico de porcentajes de los vehículos utilizados en Sevilla Capital. Fuente: (IMD, 2016).



La ilustración 1 nos muestra el elevado porcentaje de vehículos privados que utilizamos a la hora de desplazarnos. Esto puede provocar una congestión en las calles, lo que conlleva retrasos al trabajo, al ir a clase, asistir a citas médica, y afectan también a la calidad del transporte público, ya que tráfico privado y autobuses comparten el mismo viario. Además, una mayor congestión implica un mayor tiempo al buscar estacionamiento lo que entorpece el flujo del tráfico.

La estructura del trabajo es la siguiente. Primero, en la Sección 2, se hará una revisión teórica de las metodologías de análisis de redes complejas, incluyendo su caracterización, métricas básicas y las medidas de centralidad que se utilizarán para medir la vulnerabilidad.

En tercer lugar, se presenta una revisión de la literatura asociada a la temática del uso de medidas de redes complejas dentro del campo del transporte.

En la Sección 4, abordaremos las medidas en redes con pesos, a diferencia de las medidas topológicas. Pudiendo ser estos pesos, distancias, tiempo de viaje, número de carriles...

La Sección 5 presenta un análisis del parque automovilístico privado de la ciudad de Sevilla. Se realiza un análisis de la circulación dentro de la ciudad y se lleva a cabo una clasificación del viario principal de la ciudad.

Finalmente, en la Sección 6 se describen las herramientas utilizadas para la resolución del problema (Excel para el abastecimiento de datos, Eyd y Python) y se muestran los resultados más relevantes que se han obtenido.

2 TEORÍA DE GRAFOS

2.1 Terminología y conceptos básicos

Sea un sistema de relaciones entre distintos elementos, se denomina grafo a la representación matemática de dicho sistema mediante un esquema gráfico. Podemos expresar un grafo dirigido o no dirigido como $G = (V, A)$, donde V es el conjunto de nodos o vértices, cuya cardinalidad es n , y A es el conjunto de arcos o aristas. Cada arista $a \in A$ conecta dos vértices de V . La lista de arcos que forman el conjunto A se puede denotar como $A = \{(i_1, j_1), (i_2, j_2), \dots, (i_m, j_m)\}$ de modo que el número total de arcos se denota como m .

Los vértices pueden expresar sitios, páginas webs, personas, ciudades, intersecciones/cruces de calles... Mientras que los arcos pueden representar flujos, caudales, distancias, amistad, similitud...

Para tener una representación matricial de un grafo con frecuencia se utiliza la matriz de adyacencia A , que se expresa de la forma:

$$A_{ij} = \begin{cases} 1 & \text{si } (i, j) \in A \\ 0 & \text{si } (i, j) \notin A \end{cases} \quad (1)$$

En el caso de que los arcos de la red tengan pesos podemos dejar completamente definido el grafo incluyendo la matriz de los pesos, $W = (W_{ij})$.

Algunos conceptos básicos pero claves en teoría de redes son los siguientes:

- Un paseo (walk) es una secuencia en donde todos los nodos están conectados entre sí.
- Un camino (path) es un paseo formado por una serie de nodos, de los cuales ninguno se visita más de una vez, es decir, es un paseo en el que todos los nodos son distintos. La longitud topológica del camino se mide habitualmente como el número de aristas que contiene.
- Si en un camino el vértice inicial y final coinciden se le denomina ciclo (en redes no dirigidas un camino debe estar compuesto por al menos 3 vértices).
- Un camino geodésico entre dos nodos es un camino que conecta dichos nodos con distancia mínima entre ellos. Puede haber más de un camino geodésico.

2.2 Caracterización de redes complejas

En esta sección, se presentarán y se describirán algunas de las medidas específicas utilizadas en el análisis de redes complejas, así como los índices de vulnerabilidad empleados en el trabajo. Estas métricas se pueden dividir en globales, que abarcan el conjunto completo de la red y, en locales, centrándose en un nodo particular.

2.2.1 Métricas básicas

2.2.1.1 Densidad y grado de los nodos

Se define la densidad como la relación entre los arcos del grafo y el número de estos si todos los vértices estuvieran unidos entre sí, es decir, si fuera un grafo completo:

$$\rho = \frac{2m}{n \cdot (n-1)} \text{ (redes no dirigidas)} \quad (2)$$

$$\rho = \frac{m}{n \cdot (n-1)} \text{ (redes dirigidas)} \quad (3)$$

Una de las propiedades más importantes de un nodo es su grado k_i . En el caso de redes no dirigidas se obtiene de manera simple con la matriz de adyacencia A:

$$k_i = \sum_j A_{ij} = \sum_j A_{ji} \text{ (redes no dirigidas)} \quad (4)$$

Para los dígrafos se distingue entre el grado de entrada, k_i^+ , como el número de arcos que entran en i , y el grado de salida k_i^- , como el número de arcos que salen de i , de forma que el grado total de un nodo será:

$$k_i = k_i^+ + k_i^- \text{ (redes dirigidas)} \quad (5)$$

Entonces teniendo en cuenta lo anterior, la suma de los grados de entrada y salida de todos los nodos del grafo dirigido se corresponde con el número de arcos total de la red:

$$\sum_{i=1}^n k_i^- = \sum_{i=1}^n k_i^+ = m \quad (6)$$

Y a su vez como consecuencia se obtiene con la siguiente ecuación (7) el grado medio de una red dirigida como:

$$\langle k \rangle = \frac{1}{n} \sum_i k_i^- = \frac{1}{n} \sum_i k_i^+ = \frac{m}{n} \quad (7)$$

Así sustituyendo en (2) el grado medio obtenido en la ecuación anterior, se obtiene una nueva expresión que relaciona la densidad y el grado medio de la red dirigida:

$$\rho = \frac{m}{n \cdot (n-1)} = \frac{\langle k \rangle}{n-1} \quad (8)$$

Con lo cual se deduce que un grafo es poco denso cuando $\langle k \rangle$ crece más lentamente que n y denso, en la situación contraria, $\langle k \rangle$ crece más rápidamente que n .

2.2.1.2 Fuerza de los nodos

En las redes que no utilizan pesos en sus conexiones, estos se consideran de tipo binario, uno si están conectados los nodos y cero en caso contrario. Sin embargo, la posibilidad de otorgar pesos a los arcos permite dar una visión más real al análisis, ya que en numerosas circunstancias las medidas que no incorporan pesos en arcos o nodos resultan poco informativas. En el caso de que todos los pesos de la red tengan valor uno, denominaremos a la red como topológica.

La matriz de pesos W , se trata de una matriz de igual tamaño y aspecto a la de adyacencia A , salvo que los valores no son ceros y uno, sino los pesos de los arcos. Por tanto, se denomina fuerza de un nodo en redes no dirigidas a la suma de los pesos de los arcos que inciden o salen del nodo i :

$$S_i = W_{ij} = W_{ji} \quad (9)$$

Mientras que para el caso de redes dirigidas hay que diferenciar entre los pesos de los arcos que inciden en i , S^+ , y los que pesos de los arcos que parten de i , S^- . Obteniéndose la fuerza del nodo como la suma de la fuerza entrada y la fuerza de salida:

$$S^+ = \sum W_{ij} \quad (10)$$

$$S^- = \sum W_{ji} \quad (11)$$

$$S_i = S^+ + S^- \quad (12)$$

2.2.1.3 Longitud del camino medio, diámetro y eficiencia

Sabiendo de las definiciones anteriores que la distancia geodésica es el camino más corto entre dos nodos, se define la longitud del camino medio como la longitud media de los caminos geodésicos más cortos entre cualquier par de nodos:

$$\langle d \rangle = \frac{1}{n \cdot (n-1)/2} \sum_{j>1} d_{ij} \text{ (redes no dirigidas)} \quad (13)$$

$$\langle d \rangle = \frac{1}{n \cdot (n-1)} \sum_{j \neq 1} d_{ij} \text{ (redes dirigidas)} \quad (14)$$

Por otro lado, el valor máximo de d_{ij} se le conoce como el diámetro de la red:

$$D = \max d_{ij} \text{ cuando } i > j \text{ (redes no dirigidas)} \quad (15)$$

$$D = \max d_{ij} \text{ cuando } i \neq j \text{ (redes dirigidas)} \quad (16)$$

Sí dos nodos no están conectados diremos que la distancia geodésica es infinita. Generalmente para redes en las cuales queremos tener en cuenta las distancias se usa como pesos la longitud de los arcos. Esto es aplicable tanto para redes dirigidas como para las no dirigidas. Teniendo esto

en cuenta y que considerando el intercambio de información se hace utilizando las rutas geodésicas (Latora, 2001), calcula la eficiencia de transmisión global de la red queda como:

$$\eta = \frac{1}{n \cdot (n-1)/2} \sum_{j>1} d_{ij}^{-1} \text{ (redes no dirigidas)} \quad (17)$$

$$\eta = \frac{1}{n \cdot (n-1)} \sum_{j \neq 1} d_{ij}^{-1} \text{ (redes dirigidas)} \quad (18)$$

Lo que nos dice que una mayor distancia geodésica entre los nodos se traduce en una menor eficiencia de la red.

2.2.1.4 Índice de centralidad

Este índice se basa en los grados de los nodos de la red. Mide cuándo un nodo o grupos de nodos tiene más conexiones que otros. El índice se obtiene para redes no dirigidas con la siguiente expresión:

$$CI = \frac{1}{(n-1) \cdot (n-2)} \sum_i (k_{max} - k_i) \quad (19)$$

Mientras que en el caso de redes dirigidas se hace diferencia entre el índice de centralidad de entrada (CI^+) y de salida (CI^-).

$$CI^+ = \frac{1}{(n-1) \cdot (n-1)} \sum_i (k_{max}^+ - k_i^+) \quad (20)$$

$$CI^- = \frac{1}{(n-1) \cdot (n-1)} \sum_i (k_{max}^- - k_i^-) \quad (21)$$

$$CI^t = \frac{1}{2(n-1) \cdot (n-2)} \sum_i (k_{max} - k_i) \quad (22)$$

2.2.1.5 Clustering

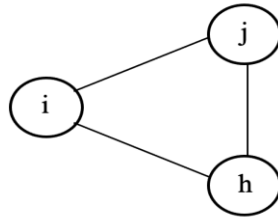
Es una propiedad típica de redes, en la que es probable que dos individuos con un amigo en común se conozcan (Boccaletti, Latora, & Chavez, 2006). Los algoritmos de clustering buscan identificar patrones y estructuras en los datos de la red, agrupando nodos que están más relacionados entre sí y separando aquellos que están menos conectados. En términos de un grafo, clustering, significa la presencia de un elevado número de triángulos que conectan nodos de la

red, así, por ejemplo, si los vecinos j y h del nodo i se conectan entre sí, se forma un triángulo en el que los tres nodos están interrelacionados (Wu C. , 2005).

$$CC_i = \frac{t_i}{T_i} = \frac{2}{k_i(k_i - 1)} \sum_{j>h} A_{ij}A_{ih}A_{jh} = \frac{1}{k_i(k_i - 1)} \sum_{j,h} A_{ij}A_{ih}A_{jh} = \frac{(A^3)_{ii}}{k_i(k_i - 1)} \quad (23)$$

Donde t_i es el número de triángulos en los que participa el nodo i y T_i el número de triángulos totales en los que podría participar el nodo i .

Ilustración 2 Formación de triángulos en red no dirigida. Fuente: Elaboración propia.



Para el caso de redes no dirigidas con pesos tenemos (Barrat, Bathelemy, & Vespignani, 2004):

$$CC_i^w = \frac{1}{S_i(k_i - 1)} \sum_{j,h} \frac{W_{ij} + W_{ih}}{2} A_{ij}A_{ih}A_{jh} \quad (24)$$

CC_i^w cuenta, para cada triángulo formado en la vecindad del nodo i , el peso de los arcos que conectan al vértice i con los otros dos nodos del triángulo. Así estamos considerando no solo el número de triángulos, sino también su peso relativo total con respecto a la fuerza del vértice (se considera fuerza a la suma de los pesos de los arcos involucrados). El factor de normalización $S_i(k_i - 1)$ tiene en cuenta el peso de cada arco multiplicado por el número de triángulos en los que puede participar, y asegurando $0 \leq CC_i^w \leq 1$.

A partir de esta medida local, el clustering de la red se puede obtener realizando un promedio de los coeficientes de clustering de los nodos integrantes:

$$CC = \frac{1}{n} \sum_i CC_i \quad (25)$$

En el caso de redes no dirigidas con pesos se utiliza:

$$CC^w = \frac{1}{n} \sum_i CC_i^w \quad (26)$$

Para redes dirigidas se usarían métricas similares, pero teniendo en cuenta las conexiones entrantes y salientes por separado.

2.2.1.6 Otras métricas globales

- Reciprocidad: se define como la propabilidad de que un arco dirigido exista en los dos sentidos, es decir, que existan conexiones mutuas entre dos nodos en una red. En términos generales también se puede definir como la fracción de arcos que forman parte de una pareja de arcos recíprocos, por tanto, se calcularía:

$$R = \frac{1}{m} \sum_{i,j} A_{ij}A_{ji} = \frac{1}{m} Tr(A^2) \quad (26)$$

Donde $A = (A_i)$ es la matriz de adyacencia binaria cuyos elementos indican si cada arco (i, j) existe o no. Esta métrica indica el grado de bidireccionalidad de las conexiones. En una red no dirigida el grado de reciprocidad vale uno. En una red dirigida el grado será cero en caso de que no haya ningún arco bidireccional.

Las redes con alta reciprocidad tienden a tener relaciones más fuertes y equitativas entre sus nodos. Por otro lado, una reciprocidad baja nos indica relaciones más asimétricas y que los nodos comparten la información de forma menos igualitaria.

- Transitividad: concepto muy similar al clustering, ya que la transitividad mide la probabilidad de que dos vecinos de un nodo estén conectados, pero calculado de manera global como la fracción de triángulos conectados en la red. Teniendo en cuenta que h, j, i pertenecen al mismo conjunto de nodos M se define:

$$T = \frac{3 \sum_{h>j>i} A_{ij}A_{ih}A_{jh}}{\sum_{h>j>i} (A_{ij}A_{ih} + A_{ji}A_{jh} + A_{hi}A_{hj})} \quad (27)$$

La transitividad es un concepto importante en el análisis de redes por que proporciona información sobre la organización y la dinámica de las relaciones entre los nodos. Una alta transitividad nos dice que existen comunidades o grupos de nodos intensamente conectados. Mientras que una baja transitividad muestra la tendencia de los nodos a ser independientes.

2.3 Medidas de centralidad

Para el análisis de la vulnerabilidad usaremos una serie de índices o medidas que se basan en la definición de centralidad dentro de la teoría de redes complejas. El objetivo es evaluar la importancia de los elementos que componen la red sabiendo que los nodos ejercen influencias

unos sobre otros a través de sus conexiones. Hay diversas formas de evaluar esas conexiones. Las medidas de centralidad se pueden clasificar en tres grandes grupos basándonos en la distinción realizada por (Boldi & Vigna, 2013). Dichos grupos o categorías se encuentran en la [Tabla 1](#) con algunos ejemplos.

Tabla 1 Medidas de centralidad por categorías. Fuente: (Freeman, 1977).

Categorías	Características	Ejemplo
Medidas geométricas	Clasifica un elemento midiendo la distancia geodésica entre pares de nodos	Closeness centrality
Medidas basadas en trayectorias	Clasifica un elemento en función del número de caminos más cortos que pasan por él	Betweenness centrality
Medias espectrales	Clasifica los elementos midiendo la importancia de sus vecinos	Eigenvector Pagerank
Medidas de grado	Clasifica los elementos en función del número de conexiones que tenga un nodo	Degree centrality

- Degree centrality: mide la centralidad de cada nodo en función de la cantidad de conexiones (k_i) que tienen en la red frente al número de nodos.

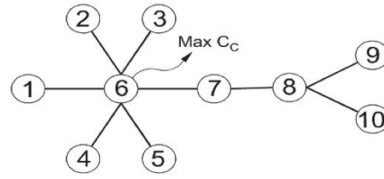
$$DC_i = \frac{k_i^+ + k_i^-}{n - 1} \quad (28)$$

- Closeness centrality: Refleja la distancia relativa entre un nodo y el centro geométrico exterior de la red. Los nodos con valor alto en este índice se encuentran cerca de otros nodos en una red. Para cuantificar la influencia de este índice se usa la expresión:

$$CC_i = \frac{n - 1}{\sum_j d_{ij}} \quad (29)$$

Donde d_{ij} es la longitud del camino más corto entre el nodo i y el nodo j .

Ilustración 3 Ejemplo de nodo con valor más alto de Closeness Centrality topológico. Fuente: (Freeman, 1977).

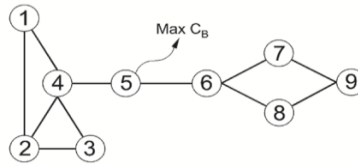


- Betweenness centrality: Los índices basados en el recuento de los caminos más cortos fueron desarrollados formalmente por (Freeman, 1977), que introdujo la betweenness centrality como una medida de probabilidad de que un camino más corto aleatorio pase por un nodo o aristas dados. En su artículo (Dong, Guan, & Xing, 2019), nos dice que la centralidad entre nodos es una medida de la frecuencia con la que un nodo determinado se encuentra en el camino más corto entre otros dos nodos. De tal forma que el índice se puede calcular como:

$$CB_i = \sum_{i,j,k \in \{1,2,\dots,n\}} \frac{d_{jk}(i)}{d_{jk}} \quad (30)$$

Donde $d_{jk}(i)$ es el número de caminos más cortos que conectan j y k a través del nodo i , y d_{jk} es el número total de caminos más cortos que conectan j y k . Un nodo con alto valor en betweenness esta en posición de amenazar a la red si interrumpe sus operaciones.

Ilustración 4 Ejemplo de nodo con el valor más alto de Betweenness Centrality topológico. Fuente: (Freeman, 1977)

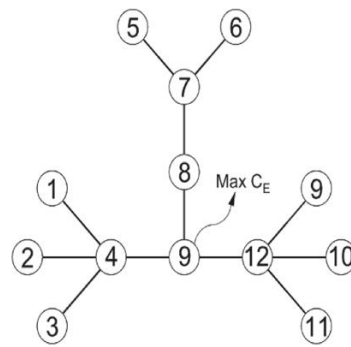


- Eigenvector centrality: Este índice calcula la importancia de un nodo basándose en los valores de centralidad combinados de sus vecinos (Zarghami & Dumark, 2021). En otras palabras, un nodo consigue un alto valor de centralidad de eigenvector (autovector) si esta conectado a muchos otros nodos que también tienen un valor alto de este índice. Así un nodo con pocas conexiones podría tener valor alto si esas pocas conexiones están muy bien conectadas con otros nodos. Se puede expresar de la siguiente manera:

$$CE_i = \frac{1}{\lambda} \sum_{j \in M(i)} CE(j) \quad (31)$$

Donde $M(i)$ representa un conjunto de los vecinos de i y λ es un valor constante. Si se reordena la expresión anterior utilizando la matriz de adyacencia, la ecuación (31) puede reescribirse como $\lambda e = Ae$, donde A es la matriz de adyacencia con el mayor valor del vector propio de λ , y e denota la centralidad de eigenvector de todos los nodos.

Ilustración 5 Ejemplo de nodo con el valor más alto de Eigenvector Centrality topológico. Fuente: (Freeman, 1977).



- PageRank: Algoritmo que, para redes dirigidas, con o sin atributos, asigna la importancia de los nodos a partir de su relación con el resto de los nodos de la red. Utiliza el concepto de grado de salida para dividir los nodos en dos grupos, aquel en el que los nodos tienen un grado de salida mayor que cero y los que tienen un grado de salida igual a cero, que se denominan nodos terminales (Brin & Page, 1998).

El objetivo es asignar una mayor importancia a los nodos que reciben un mayor número de conexiones entrantes. El algoritmo de cálculo del PageRank formula que la importancia de un nodo se distribuye en su mayoría en función de su grado de salida cuando es mayor que cero y en menor medida entre los nodos con un grado de salida igual a cero, asignando un valor residual a los nodos terminales. Los nodos con mayor valor en PageRank se consideran de mayor importancia dentro de la red. Todo este proceso depende principalmente de tres parámetros:

1. d : (dumping factor) valor comprendido entre cero y uno. Generalmente se toma como 0.85. Se utiliza para evitar problemas de convergencia.
2. $C(T_i)$: número de enlaces salientes del nodo i .
3. $PR(T_i)$: valor PageRank del nodo i conectado al nodo A .
4. N : número total de nodos en la red.

$$PR(A) = \frac{(1 - d)}{N} + d \sum_i \frac{PR(T_i)}{C(T_i)} \quad (32)$$

3 REVISIÓN DE LA LITERATURA

La teoría de redes complejas es una herramienta útil para analizar los sistemas de transporte y comprender sus patrones de comportamiento, eficiencia y vulnerabilidad. La aplicación de esta teoría a las redes de transporte ha generado investigaciones en diferentes contextos, incluyendo carreteras, ferrocarriles, transporte aéreo, transporte público... A continuación, se presentan algunas áreas claves y resultados de investigación en este campo.

3.1 Concepto de fiabilidad

A pesar de que los conceptos de fiabilidad y vulnerabilidad están estrechamente relacionados, la literatura sobre la fiabilidad en redes es menos abundante que la literatura sobre la vulnerabilidad, pero la fiabilidad ofrece una visibilidad más amplia en cuanto a mantener o recuperar la funcionalidad de una red frente a una interrupción de esta.

(Holling, 1973) introduce el concepto en la descripción de ecosistemas y la estabilidad de la evolución ecológica, mostrando la capacidad del sistema para resistir shocks y mantener el equilibrio. (Manikan, 2017) define la fiabilidad como la capacidad intrínseca de un sistema para ajustar su funcionamiento antes, durante, o después de cambios y perturbaciones de modo que pueda sostener las operaciones requeridas bajo condiciones tanto esperadas como inesperadas.

Desde entonces este concepto se ha ido ampliando a otros campos como la economía y la ingeniería, en las cuales se incluye la fiabilidad del transporte.

3.2 Medidas en las que basar la fiabilidad de una red de transporte

Conviene disponer de medidas de la fiabilidad de una red de transporte debido principalmente a que las sociedades están cada vez más interconectadas y, por tanto, la mejora de movilidad resulta también fundamental para la sociedad y la economía.

(Cox & Rose, 2011) desarrolla métricas para la fiabilidad de los sistemas de transporte de pasajeros ante el terrorismo. Sus medidas de fiabilidad son específicas para el número de viajes. En su estudio sobre los atentados en el metro y autobús de Londres en julio de 2005, encuentran que el 77.4% de la reducción total de viajes fue compensada por aumentos en modos alternativos durante los cuatro meses siguientes a los ataques. Analizan datos empíricos como el porcentaje de interrupción de viajes evitados por comportamientos de fiabilidad. Los comportamientos sobre fiabilidad se refieren al aumento de modos alternativos ya que examinan datos sobre vehículos y desplazamientos caminando durante dicho periodo. Esto ilustra la capacidad del sistema de transporte para absorber el impacto de un ataque terrorista. Por otro lado, (Allenby & Fink, 2005) señala que cada sistema se enfrenta a un conjunto único de desafíos. Los desafíos únicos combinados con las diferencias entre infraestructuras y el flujo del tráfico, conducen a la incapacidad de transferir técnicas de análisis. Sin embargo, asegura que algunas medidas como el tiempo de viaje, la calidad del servicio y el tiempo de recuperación si son aplicables. En

opinión de los autores, para que las investigaciones se apliquen más a la práctica, es necesario la colaboración global entre las autoridades responsables y otros interesados para el aprendizaje mutuo y la transferencia de conocimientos.

En su trabajo Nicholson (Nicholson, 2007), centrado en la reducción de impactos provocados por desastres naturales (tsunamis y terremotos), utiliza categorías para reducir la falta de fiabilidad de la red de carreteras según cuándo ocurran:

- Reducción (es decir, identificar y reducir riesgos mediante mejoras en la infraestructura).
- Preparación (es decir, capacitar al personal de defensa civil e instalar sistemas de alerta).
- Respuesta (es decir, reaccionar ante situaciones de emergencia a corto plazo).
- Recuperación (es decir, minimizar los impactos socioeconómicos a largo plazo).

Seguidamente describe métodos para evaluar y mitigar los impactos de la degradación de las redes viales interurbanas y urbanas. Señala como posibles opciones remplazar o fortalecer puentes, construir nuevos enlaces, disponer de más opciones de servicios de emergencias poniendo de ejemplo puentes Bailey y monitorear puntos críticos entre otras alternativas.

3.3 Vulnerabilidad de las redes de transporte

(Lars-Göran & Jenelius, 2015) en su artículo propone una definición de vulnerabilidad válida para diferentes modos de transporte, “La vulnerabilidad en el sistema de transporte por carretera es la susceptibilidad a incidentes que pueden resultar en daños considerables”. Mientras que (Kaplan & Garrick, 1981) sugiere que el análisis del riesgo debe responder a las preguntas: “¿Qué puede pasar?, ¿Qué probabilidad hay de que ocurra? y, ¿Cuáles son sus consecuencias?” para cada escenario de riesgo concebible. Esta definición de vulnerabilidad es más rica puesto que considera no solo el posible incidente si no cuáles son los problemas que traería consigo.

También (Lars-Göran & Jenelius, 2015), exponen dos formas de analizar la vulnerabilidad. Estudiando la topología de la red, si es dirigida o no dirigida, si está ponderada o no, y realizando un análisis basado en el estudio del sistema, es decir, modelando la demanda de viaje en términos de generación y atracción de viajes, elección de destino y de modo, y determinando la elección de rutas como un problema de equilibrio en la red teniendo en cuenta la interacción con la congestión y los retrasos.

Por otro lado, (Appert & Chapelon, 2013), llevan a cabo una distinción en la definición de la vulnerabilidad. En el artículo consideran que los incidentes pueden ser más o menos predecible, causados voluntaria o involuntariamente, por el hombre o por la naturaleza. No consideran riesgo el cierre por mantenimiento y obras viales ya que los conductores pueden conocer esos incidentes con anticipación y posponer o desviar sus viajes en consecuencia. Por tanto, ellos llegan a la conclusión de que los bloqueos y problemas de congestión a considerar deben ser debidos a situaciones aleatorias, y no planificadas, sobre el flujo del tráfico.

3.4 Vulnerabilidad de redes por carreteras urbanas

En esta categoría se incluyen algunos trabajos que se pueden utilizar para orientar a las administraciones de carreteras a la hora de priorizar su mantenimiento y la reparación. Las redes viales son infraestructuras críticas debido a que el transporte por carretera abarca la logística, los viajes de negocios y los desplazamientos hacia el trabajo entre otros muchos otros

desplazamientos. El estudio de la red de carreteras del norte de Suecia de (Olsson, 2005) utiliza el tiempo de viaje entre los nodos como coste generalizado de viaje y el tiempo de viaje lo calcula dividiendo la distancia por la velocidad del flujo libre asumiendo que los tiempos de viaje son independientes a la carga de tráfico, lo que permite utilizar algoritmos más simples y rápidos.

Por otro lado (Wang, Pei, & Liu, 2023) estudian la vulnerabilidad de las redes de carreteras después de un fallo desde la perspectiva de la topología de la red, que hace que la red conectada se divida en múltiples subredes. Estudian la vulnerabilidad con la nueva densidad vial urbana, los nuevos tiempos promedio de viaje de la red y centrando importancia en la subred con mayor número de nodos. (Yingying & Feng, 2013) utilizan un enfoque similar, pero, los ataques para provocar fallos los generan de dos formas, primero de manera aleatoria y, en segundo lugar, de forma intencionada para evaluar la supervivencia de la red en diferentes situaciones. (Sunghoon & Hwasoo, 2016) centran su estudio en la ciudad de Seúl proponiendo un índice de vulnerabilidad basado en el flujo para redes viales urbanas, utilizando el diagrama fundamental macroscópico (MFD). El MFD relaciona el flujo medio espacial y la acumulación de vehículos en una red vial. Y el índice les sirve para comparar el comportamiento del MFD en condiciones normales y cuando uno o más enlaces de la red fallan. Este estudio permite priorizar la forma de ejecutar las acciones de reparación durante un día.

En (Leal De Oliveira & Da Silva Portugal, 2014) ponen el foco en la región metropolitana de Río de Janeiro. Basan su estudio sobre la vulnerabilidad en índices de congestión para los enlaces de la red, que clasifican en tres grandes grupos. Aquellos que evalúan la congestión en función de como afecta el flujo del tráfico al enlace bajo análisis, para los que utilizan el cociente V/C (volumen/capacidad). Al segundo grupo pertenecen aquellos que evalúan el flujo en términos de velocidad promedio o tiempo promedio y, dentro de este grupo, utiliza el TTI (cociente entre el tiempo de viaje promedio en condiciones de congestión y el tiempo de viaje promedio en condiciones de flujo libre). El tercer grupo está compuesto por índices que no puede añadir en los dos grupos anteriores. La finalidad del análisis es la de realizar una jerarquía multicriterio para la planificación de acciones correctoras e inversiones en los enlaces de la red de carreteras.

(Kun, Wei, & Xinram, 2022) suponen una ciudad dividida por un río, lo que se asemeja al caso de Sevilla. Modelan el problema de vulnerabilidad como un problema no lineal mixto-entero (MINLP). El nivel superior busca la combinación de los enlaces más críticos para medir el impacto en toda la red, mientras que el nivel inferior, guía la distribución de los flujos de tráfico dado un escenario de interrupción particular. Llegando a la conclusión de que la combinación de carreteras críticas no es el grupo de carreteras que funcionan peor en problemas críticos individuales. Otro estudio que centra su análisis en el efecto de colapso en un puente es el de, en (Feng & Levinson, 2011) concreto el puente I-35W que cruza el río Misisipi. El colapso de dicho puente causó pérdidas diarias de 71000 a 220000 dolares, dependiendo de las respuestas de los usuarios a la hora de ajustar sus destinos debido al cierre del puente. El estudio utiliza un modelo de demanda de viajes para realizar un análisis rápido en situaciones de emergencia.

Se encuentran muchos artículos sobre la vulnerabilidad de la red de transporte urbana ante inundaciones, fenómenos que sufren algunas ciudades que parecen ser consecuencias del cambio climático. Un ejemplo es el trabajo de (Borgato Morelli & Luiz Cunha, 2021), el cual se focaliza en la ciudad de São Carlos en Brasil. Los resultados de sus análisis basados en dos conjuntos de datos (la estructura de la red y la distribución de los viajes), muestran que una planificación urbana compacta que fomente caminar o andar en bicicleta en distancias relativamente cortas, puede ser mas resistente a las inundaciones que un plan que promueve el desplazamiento individual motorizado.

Otro estudio sobre redes de transporte e inundaciones es el propuesto por (Jie & Zhenwu, 2021). Definen y miden la vulnerabilidad de la red vial en función de la exposición del área propensa a inundaciones y la importancia de dicha área. Estudian las áreas de la ciudad de Wuhan propensas a inundaciones y toman los datos sobre el tráfico en condiciones normales y datos sobre el tráfico afectado por inundaciones. Los hallazgos fueron varios, el primero que la mayoría de las áreas propensas se encuentran cerca de ríos y lagos, en segundo lugar, que durante un estado afectado los usuarios eligen el transporte público en lugar de vehículos privados y, por último, que las propiedades topológicas de las áreas propensas a inundaciones influyen significativamente.

3.5 Redes de Transporte público

En el ámbito de las redes de transporte público (PTN) hay dos tipos principales de estudios, en el primero grupo encontraremos estudios que se basan en un análisis puramente topológico de la red, mientras que, en el segundo grupo, más frecuente, no sólo usan aspectos de la topología de la red, sino que se incorporan datos de tránsito para poder valor en mayor profundidad los efectos económicos. El trabajo de (Mcleod, Vecsler, & Yuan, 2017) se incluye dentro de la segunda opción, puesto que analizan el sistema de metro de la ciudad de Nueva York basándose en el horario de servicio, el volumen de pasajeros y los datos origen-destino.

Con el objetivo de reducir emisiones, ahorrar energía y aliviar la congestión del tráfico (Wang & Zhang, 2019) analiza la conectividad del metro de Najing simulando ataques a los nodos de mayor grado, y llega a la conclusión de que el aumento en el coste de construcción tiene un impacto en la solidez de las redes de transporte ferroviario. En el trabajo de (Huang & Zhang, 2022) se comparan cinco ciudades húngaras con parecidos tanto en la red de transporte público (autobús, tranvía) como en características geográficas. Realizan un análisis utilizando varias medidas de centralidad, Closeness centrality, Betweenness centrality y Pagerank para ver cuales son los enlaces (rutas) y nodos (estaciones) más sensibles de la red, de forma ponderada y no ponderada.

(Rodríguez Nuñez & García Palomares, 2014) llevan a cabo un estudio de la vulnerabilidad de la red de metro de Madrid. La metodología consiste en empezar obteniendo las rutas de viaje óptimas entre cada par de estaciones, se combinan con el número total de viajes entre estaciones y se calcula la suma del número de viajes en todas las rutas óptimas que pasan por cada enlace de la red. De esta forma es posible medir las consecuencias del cierre de un enlace tanto si hay alternativa de viaje como si no la hay. Finalmente llegan a la conclusión de la importancia de las líneas circulares como elemento de robustez de la red, puesto que, al generar alternativas de viaje, permiten absorber los impactos de las incidencias en los distintos enlaces de la red.

(Wei, Wu, & Yin, 2023) realizan un estudio más amplio, puesto que eligen cuarenta ciudades para realizar sus evaluaciones en la red de transporte y posteriormente comparar los resultados. En su modelo se provocan perturbaciones en los enlaces, pudiendo ser estos accidentes de tráfico y fallos en semáforos por cortes de energía, entre otros. Estas perturbaciones se estudian como eventos aleatorios, localizados y un tercer tipo de perturbación estudiado corresponde a las inundaciones. En la mayoría de los casos obtienen que el efecto de las inundaciones es el menos disruptivo dentro de los tres tipos de interrupciones analizados.

(Cats & Jenelius, 2015) proponen una metodología para evaluar la efectividad de un aumento estratégico de la capacidad en enlaces alternativos de la PTN, con el objetivo de reducir el impacto de interrupciones inesperados en la red. Esto requiere un procedimiento iterativo para

identificar enlaces sobresaturados con el fin de tener en cuenta la interacción entre la capacidad del servicio, las interrupciones y los flujos de los pasajeros. El estudio se aplicó para el PTN de Estocolmo. El trabajo de (Gedik, Medal, & Rainwater, 2014), se centra en una red ferroviaria. Los autores eligen una cantidad limitada de elementos de la red que son atacados y seguidamente difieren los flujos de peatones a trenes en la red residual, creando un escenario que les permite realizar una evaluación de la vulnerabilidad, incluyendo decisiones de desvío, costes de los nuevos viajes y tiempos de demora. Sus resultados muestran que el aumento del número de interrupciones no tiene impacto en el coste del transporte, mientras que el coste de retraso debido a las congestiones en la búsqueda de nuevas rutas para los trenes aumenta drásticamente.

(Sun & Zhao, 2015) también se centra en el transporte ferroviario y su importancia en la vulnerabilidad de una PTN. Estudian la vulnerabilidad del metro de Shanghai utilizando teoría de redes complejas y grafos, y proponen un modelo que tiene en cuenta factores como la interconexión de las líneas de metro y el flujo de pasajeros. Como conclusión obtienen que la red ferroviaria es vulnerable en ataques a los nodos que disponen de un mayor número de conexiones a otros nodos, que no necesariamente tienen que ser las ubicaciones con mayor flujo de pasajeros.

3.6 Relación entre la topología de la red y el flujo del tráfico

Numerosos estudios muestran la existencia de una estrecha relación entre la capacidad de tráfico de la red y la topología de esta. Redes con topologías razonables pueden evitar de forma más efectiva problemas de congestión. Muchas redes muestran características de comunidad, por lo que es de gran importancia estudiar la influencia de la estructura de comunidad en la capacidad del tráfico de la red.

En (Ma & Wang, 2023) se estudia la influencia de la estructura de comunidad de la red en la capacidad del tráfico, reduciendo el número de comunidades, aumentando el grado promedio (disponer de rutas alternativas) y la fuerza de conexión de la comunidad. Como resultado se obtiene que aumenta la capacidad del tráfico debido a que la carga por nodo disminuye, lo que facilita al sistema el manejo de flujo de vehículos, pero al mismo tiempo, el número de rutas alternativas crece, y aunque la capacidad global del sistema mejora, la mayor variedad de rutas genera desafíos en la gestión del tráfico pudiendo complicar la regulación de este.

Para (Wu & Hu, 2021) también están relacionados los dos conceptos en cuestión. Considera que hay dos indicadores principales para analizar el comportamiento del tráfico dentro de la estructura de la red, la tasa máxima de llegada (la tasa de finalización de los viajes) y, la densidad de congestión. En la simulación realizada para su estudio, se basan en una red rectangular ($N \times M$) y van variando la relación $r = N/M$. Utilizan dos estrategias de enrutamiento, ruta geográfica más corta y ruta temporal más corta. Se concluye que una configuración de la red con carreteras más alargadas retrasa la tasa de llegada y aumenta la congestión de vehículos. Este fenómeno se explica debido a la distribución y necesidad de giro de los vehículos y, a la distancia de viaje.

4 MEDIDAS EN REDES CON PESOS

En este capítulo vamos a volver a tratar las medidas de centralidad, pero en este caso teniendo en cuenta la inclusión de pesos que permitan realizar un análisis en mayor profundidad, teniendo en cuenta que los arcos y nodos de la red pueden pertenecer a diferentes categorías. No basta con medir la red de manera topológica, precisamos de más información para hacer un estudio más rico y que se asemeje más a la realidad en la medida de lo posible.

4.1 Medidas de centralidad en redes urbanas con datos

En el contexto de la ciudad es importante entender los conceptos de centro y centralidad. A lo largo de la historia, el centro solía ser donde se concentraban instituciones y poder. En las ciudades de hoy en día, se encuentran múltiples centros especializados, comerciales, culturales, administrativos, etc.

La centralidad urbana se refiere a la capacidad de un lugar de ser un centro referente y ser reconocido socialmente como punto de confluencia. La centralidad en redes complejas depende de la topología de las relaciones entre sus nodos, pero en las ciudades no solo se basa en la topología, sino también en las actividades que se desarrollan en esos lugares. Para entender mejor el rol que juegan los diferentes lugares importantes de una ciudad, se deben considerar datos geolocalizados que describen las actividades junto con la topología urbana.

Anteriormente, en la sección 2, se describieron las medidas de centralidad sin atributos. Útiles para análisis topológicos, es decir, análisis en los que se consideran que las longitudes de los arcos de la red toman valor a uno. A continuación, se presentan los atributos que hacen posible estas consideraciones. Las centralidades ahora permiten la incorporación lógica y significativa de datos para el análisis urbano.

- **Closeness Centrality:** En su versión ponderada puede tener en cuenta factores como el tiempo de viaje, la distancia real o el coste para moverse entre los nodos. El tiempo de viaje es difícil de reflejar de manera exacta, pero conociendo la velocidad a la que se puede circular por las vías y la distancia entre nodos, se puede obtener una estimación del tiempo empleado de un nodo a otro.
- **Betweenness Centrality:** Se pueden utilizar como pesos la cantidad de recursos o información, en nuestro caso vehículos, que circulan por la vía o que atraviesan un nodo. Teniendo en cuenta la capacidad de la vía, por ejemplo, considerando el número de carriles.
- **PageRank:** Para este medidor de centralidad los atributos a tener en cuenta para medir la importancia de los nodos podrían ser, el ancho de la vía, ya que permitiría manejar más flujo de tráfico, el número de carriles, relacionado con el anterior, al ser indicativo del flujo de tráfico, y, por último, los puntos de interés cercanos, hospitales, escuelas, centros comerciales... Este último es el más complicado de implementar debido al tamaño de la red que se usa en la parte experimental.

En la ilustración 6 se muestra la red de estudio clasificando los arcos según el número de carriles que disponen para el soporte del flujo de vehículos. Los arcos de mayor grosor disponen de un número mayor de carriles que los arcos de menor grosor. Esta ilustración permite deducir los posibles arcos que podrían ser más vulnerables de la red.

Ilustración 6 Mapa de estudio con carriles como atributos. Fuente: Elaboración propia.



Mientras que en la ilustración 7, se refleja la red clasificando los arcos según la velocidad a la que pueden circular los vehículos por el arco en el que se encuentran. Se diferencian tres velocidades diferentes, cuarenta, cincuenta y ochenta kilómetros por hora. Tanto las ilustraciones 6 como la 7 se han creado mediante el algoritmo del Programa 2.

Ilustración 7 Mapa de estudio con velocidades como atributos. Fuente: Elaboración propia.



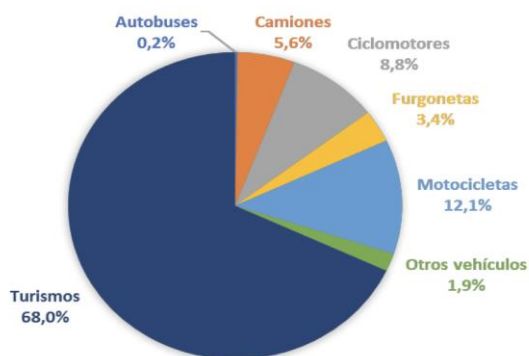
5 APLICACIÓN DE RED DE VEHÍCULO PRIVADO DE LA CIUDAD DE SEVILLA

5.1 Parque automovilístico

Según la Dirección General de Tráfico, el parque de vehículos de Sevilla cuenta con 471.126 de unidades privadas, de las cuales el 45% pertenecen al conjunto urbano. Existen 679 vehículos por cada mil habitantes.

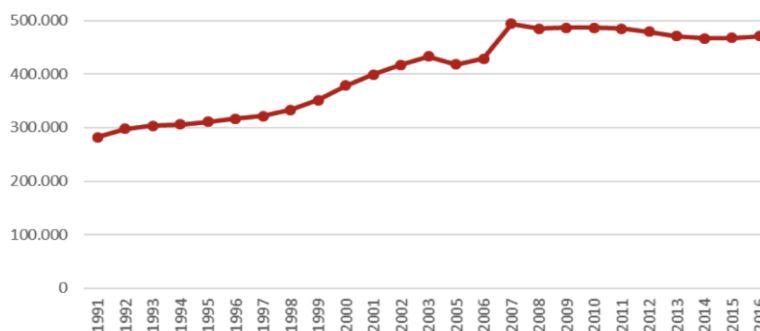
Los automóviles representan más de la mitad del parque de los vehículos del municipio, seguidos en orden de importancia, de las motocicletas y los ciclomotores (ver Ilustración 8).

Ilustración 8 Porcentajes de los diferentes tipos de vehículos en la ciudad de Sevilla. Fuente: (IMD, 2016).



Aunque la población de Sevilla y el parque de vehículos representan porcentajes similares, cabe señalar que en los últimos años la población ha decrecido ligeramente, mientras que el parque de vehículos ha crecido de manera significativa, como se puede apreciar en la Ilustración 9.

Ilustración 9 Evolución del parque de vehículos de Sevilla. Fuente: (IMD, 2016).



5.2 Movilidad urbana

La movilidad varía a lo largo del día, presentando diferentes periodos de hora punta. En Sevilla claramente la hora punta de mañana corresponde con el intervalo entre las 8:00 y las 9:00 horas y la del mediodía con el intervalo entre 14:00 y las 15:00 horas.

El principal motivo de los viajes es el traslado al domicilio, ya sea como origen o como destino, seguido de desplazamientos por motivo de trabajo, ocio y compras. Las tres zonas que más viajes generan ordenadas, ordenadas de mayor a menor, son: Sevilla Este/ Torreblanca/Alcosa, Cerro-Amate y en tercer lugar la Macarena, que corresponden con las zonas con más residentes. Por otro lado, las zonas que más viajes atraen se corresponden con el Casco Antiguo, seguido de Nervión y el Aljarafe, también se puede incluir la zona sur que incluye Hytasa, el aeropuerto y Su Eminencia.

La evolución de la movilidad en las últimas décadas viene marcada por un pronunciado aumento del grado de motorización de la población y un incremento de las necesidades de desplazarse provocado por la expansión de la ciudad, el crecimiento incontrolado de los núcleos localizados en los alrededores y la creación de espacios urbanos diferenciados de trabajo y residencia.

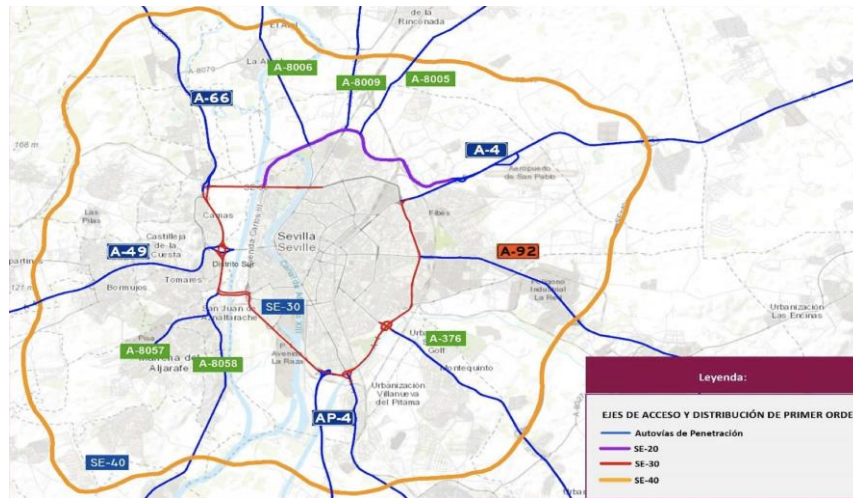
La ilustración 10 muestra las once macrozonas en las que se divide el municipio de Sevilla. Cabe destacar la división del distrito de Triana en la zona de La Cartuja y el barrio de Triana. La Cartuja destaca por ser destino de lugar de trabajo, pero su población representativa es tan pequeña que los datos no son significativos.

Ilustración 10 Distritos de Sevilla. Fuente: (IMD, 2016).



La Ilustración 11 representa los ejes y rondas exteriores que permiten la entrada y salida al municipio de Sevilla. La SE-40 y las autovías de penetración se salen del dominio de estudio, pero influyen claramente en el flujo que reciben la SE-30 y la SE-20.

Ilustración 11 Ejes y Rondas Exteriores. Fuente: (IMD, 2016).



5.3 Viario

El viario de una ciudad está estrechamente ligado a la circulación en la misma. La red de Sevilla constituye el conjunto de vías existentes que permiten la movilidad de las personas y vehículos a través de la ciudad.

5.3.1 Jerarquización del viario

Consiste en establecer una clasificación de todas sus calles para que se pueda gestionar la movilidad adecuadamente. La propuesta de jerarquía cumple con un conjunto de criterios de valoración tales como su condición funcional, la composición del tráfico y la magnitud del tráfico que circulan por ellas. Con esta clasificación se puede gestionar la movilidad de la ciudad definiendo qué papel tendrá cada calle dentro de la red urbana. Por tanto, atendiendo a los criterios anteriores de funcionalidad y movilidad, el viario se puede clasificar en los siguientes grupos:

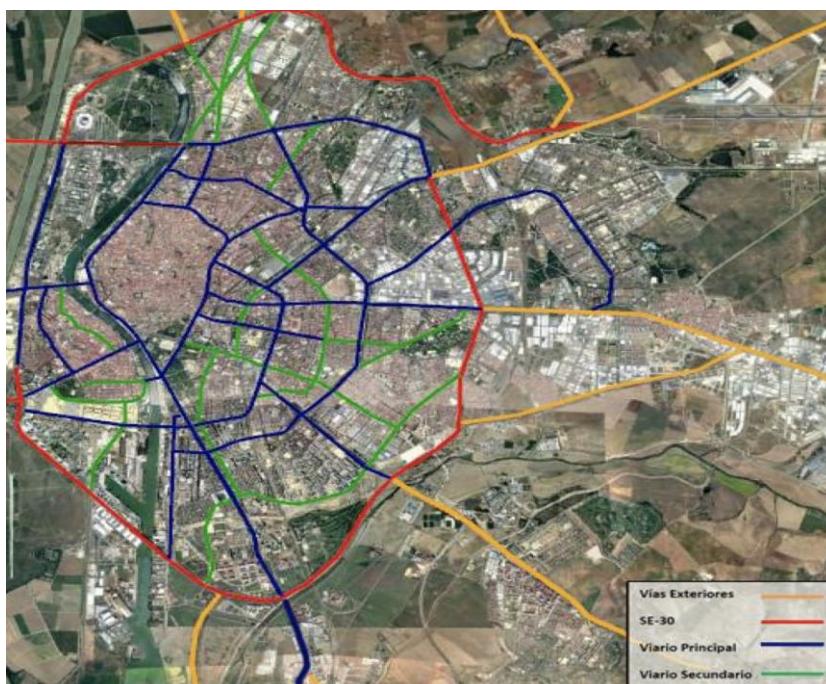
- Ejes exteriores: Se consideran en la ciudad de Sevilla como las vías radiales de alta capacidad formadas por los accesos de primer orden de la ciudad. Accesos de entrada y salida en sentido Huelva, acceso norte, Córdoba, Málaga, Utrera, Cádiz, y el acceso de la zona del aljarafe. No se entra en más detalles porque estas vías quedan fuera del dominio de estudio.
- Rondas exteriores: Son las vías de circunvalación de la ciudad que generan un efecto barrera entre el ámbito urbano e interurbano. Son el punto de confluencia de la mayoría de los ejes exteriores, siendo las primeras infraestructuras que los ejes exteriores se encuentran en su acceso a la ciudad. Las rondas exteriores corresponden a la ronda Metropolitana SE-30 y la ronda Supernorte SE-20.
- Puentes del viario exterior: La geografía de Sevilla, rodeada por el río Guadalquivir y algunos afluentes, requiere el empleo de puentes que permitan esa unión de las infraestructuras de entrada y salida de la ciudad con los ejes exteriores. Los puentes que

realizan estas funciones son cuatro, el Puente del Rey Juan Carlos I y de la Reina Sofía, Puente del Centenario, Puente del Patrocinio y el Puente del Alamillo.

- Viario principal: constituido por aquellas vías que canalizan principalmente los flujos de carácter metropolitano hacia el interior, así como parte de los desplazamientos locales de cierta distancia. Este viario compone el sistema arterial de la ciudad teniendo como función principal asegurar la movilidad motorizada dentro de la misma, articulando la ciudad en distritos y barrios. Para incluir el viario dentro de esta categoría hay que atender a estos criterios: la intensidad media diaria (vehículos/día) y, a la funcionalidad de la vía. Algunos ejemplos de viario principal pueden ser la Avenida de Kansas City, Avenida de la Paz, Avenida Carlos III, Avenida de las Razas...
- Viario secundario: Tiene la función fundamental de complementar al viario principal, de manera que ordena los distintos sectores de la ciudad y las relaciones entre ellos. Es la red que canaliza los tráficos generados/atraídos en los distintos sectores urbanos hasta el viario principal. Por otro lado, pretende favorecer los recorridos de peatones y ciclistas frente al tráfico motorizado.
- Puentes urbanos de la ciudad: Dada la situación singular de Sevilla junto con el río Guadalquivir, los puentes juegan un especial papel en cuanto a la movilidad de esta. Algunos de ellos son el Puente Cristo de la Expiración, Puente de San Telmo, Puente de la Barqueta, Puente de las Delicias, etc.

En la Ilustración 12 se representa el viario que hace posible los desplazamientos interurbanos según la clasificación anteriormente propuesta.

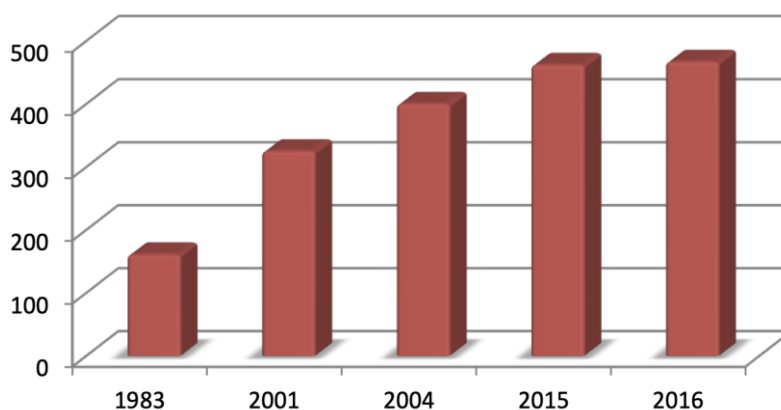
Ilustración 12 Distribución del viario principal y secundario de la ciudad. Fuente: (IMD, 2016).



5.4 Circulación

Una vez analizado el viario existente, entendido como oferta, se procede al análisis de la demanda, es decir, el tráfico que soportan los viales ya identificados. Se centrará el análisis en el tráfico interior. Un factor a tener en cuenta antes de comenzar es el índice de motorización, que muestra la tasa de vehículos en la ciudad de Sevilla, y que toma un valor de 466,9 vehículos/1000 habitantes. En la Ilustración 13 se muestra la evolución de este índice desde el año 1980.

Ilustración 13 Índice de motorización. Fuente: (IMD, 2016).



5.4.1 Análisis del tráfico interior

Los datos con los que se cuenta se han obtenidos de la infraestructura de gestión con la que trabaja la ciudad de Sevilla, el Centro de gestión de Movilidad. Este centro cuenta con un circuito cerrado de unas setenta y nueve cámaras que permiten ir observando la evolución del tráfico en tiempo real. A partir de la información recopilada se puede decir que la Ronda Urbana Norte es la vía por la cual circulan más vehículos a diario en Sevilla. Se divide en dos sectores. El que más tráfico tiene es el Este, que es el que cruza la vía del tren de Alta velocidad y se extiende hasta el nudo de la Gota de Leche.

En segundo lugar, destaca el sector oeste de la misma Ronda Urbana Norte, que corresponde a la zona de la vía próxima a Pino Montano y la glorieta de San Lázaro, justo antes de llegar a la avenida de las Juventudes Musicales, para enlazar ya con el Alamillo.

Otro de los puntos más transitados es el Puente del Cristo de la Expiración, que es la entrada y salida de aquellas personas que vienen a diario procedentes de municipios como Camas, Tomares, Bormujos, etc.

Una de las Avenidas con bastante intensidad de tráfico es la Avenida de Kansas City, que recoge la entrada de la autovía A-4.

También hay que nombrar el puente de las Delicias como tramo concurrido a tener en cuenta dentro del análisis.

Por último, en la tabla 2 se resumen las vías principales y secundarias según su frecuencia, medida en vehículos/día.

Tabla 2 Vías principales de mayor flujo. Fuente: (IMD, 2016)

Nombre de la vía	Veh/día	Jerarquización
Ronda Urbana Norte Este	84.434	Viario Principal
Patrocinio-Avenida Expo' 92	75.925	
Puente del Alamillo	63.456	
Ronda Urbana Norte Oeste	59.179	
Avenida Kansas City	58.790	
Puente de las Delicias	56.495	
Avenida de las Juventudes Musicales	54.217	
Avenida Cardenal Bueno Monreal	53.605	

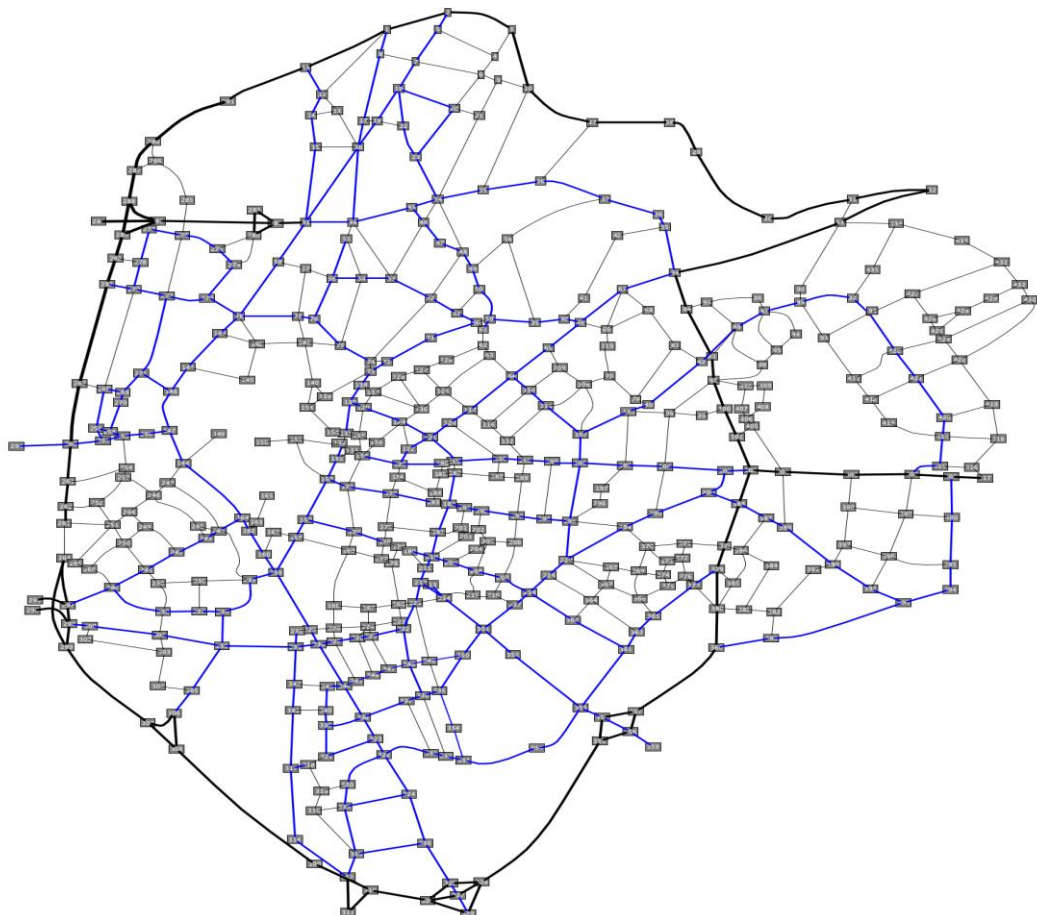
6 IMPLEMENTACIÓN Y RESULTADOS

Para nuestro estudio de vulnerabilidad utilizaremos los índices Closeness, Betweenness y PageRank. Con ellos será posible determinar saber qué nodos, cruces de calles, son los más vulnerables de la ciudad sevillana. A su vez, se ha estudiado el número de caminos mínimos que pasan por cada arco con el tiempo como atributo y el flujo de personas que circulan por cada arco y nodo de la red con el fin de conocer los arcos que hacen más vulnerables a la red viaria. Por lo tanto, a continuación, procedo a explicar el proceso que he seguido para determinar los elementos más vulnerables, las herramientas utilizadas y los resultados obtenidos.

6.1 Recopilación de datos

Para comenzar el trabajo se partió de una representación gráfica de la red de Sevilla, usando la aplicación yEd, que permite disponer de una representación gráfica de la ciudad de Sevilla, mostrando solo con los nodos y arcos a estudiar. El mapa lo componen 435 nodos y 1266 arcos. La información de las coordenadas se traslada a una hoja Excel, anotando la posición que ocupa cada nodo en el mapa para más adelante poder calcular de forma automática mediante un script de Python (salvo para arcos curvos) la distancia real en metros de cada arco, expresada en metros.

Ilustración 14 Mapa de estudio visto en la aplicacion yEd. Fuente: Elaboración propia.



Disponer del mapa en yEd también permitió etiquetar cada uno de los arcos con su correspondiente calle. De esta forma se construyó un archivo Excel con toda la información necesaria, arco a arco, el nombre de la vía, la velocidad de circulación por la vía en kilómetros hora, el número de carriles y la longitud en metros en caso de que el arco sea curvo (en otro caso se calculara automáticamente), además se indica si el arco/calle es o no bidireccional. Para conseguir todos los datos se ha utilizado como apoyo la aplicación Google Maps. En la siguiente tabla queda se muestran los primeros cien arcos a modo de ejemplo.

Tabla 3 Descripción de los arcos que componen la red. Fuente: Elaboración propia.

Origen	Destino	Factor	Tipo	Velocidad	Sentidos	Carriles	Localización
1	2	1	3	80	2	2	Rda. Norte de Sevilla
2	1	1	3	80	2	2	Rda. Norte de Sevilla
2	3	600	3	80	2	2	Rda. Norte de Sevilla
3	2	600	3	80	2	2	Rda. Norte de Sevilla
3	4	1	1	40	2	2	Ctra. Brenes
4	3	1	1	40	2	2	Ctra. Brenes
4	5	1	1	40	2	2	C/ Luna
5	4	1	1	40	2	2	C/ Luna
5	7	1	2	50	2	3	Av Biología
7	5	1	2	50	2	3	Av Biología
7	6	1	1	40	2	1	C/ Termodinámica
6	7	1	1	40	2	1	C/ Termodinámica
7	8	1	1	40	2	1	C/ Miguel Romero Martínez
8	7	1	1	40	2	1	C/ Miguel Romero Martínez
6	9	1	1	40	2	2	C/ Estrella Castor
9	6	1	1	40	2	2	C/ Estrella Castor
9	10	1	1	40	2	2	C/ Estrella Castor
10	9	1	1	40	2	2	C/ Estrella Castor
5	2	1	2	50	2	2	Av Biología
2	5	1	2	50	2	2	Av Biología
8	1	1	2	50	2	1	C/ Alcala del río
1	8	1	2	50	2	1	C/ Alcala del río
3	10	1	3	80	2	2	Rda. Norte de Sevilla
10	3	1	3	80	2	2	Rda. Norte de Sevilla
11	1	1	3	80	2	2	Rda. Norte de Sevilla
1	11	1	3	80	2	2	Rda. Norte de Sevilla
11	12	1	2	50	2	1	Ctra. de la Algaba
12	11	1	2	50	2	1	Ctra. de la Algaba
12	1	1	1	40	2	1	C/ Pez espada
1	12	1	1	40	2	1	C/ Pez espada
12	14	1	2	50	2	2	C/ José Galán Merino
14	12	1	2	50	2	2	C/ José Galán Merino
14	13	1	1	40	1	1	C/ Larache
12	13	1	1	40	2	1	C/ Navarra
13	12	1	1	40	2	1	C/ Navarra

15	14	1	2	50	2	2	C/ José Galán Merino
14	15	1	2	50	2	2	C/ José Galán Merino
16	15	1	1	40	2	1	C/ Alcalá del río
15	16	1	1	40	2	1	C/ Alcalá del río
16	13	1	1	40	2	1	C/ Navarra
13	16	1	1	40	2	1	C/ Navarra
17	16	1	2	50	2	3	C/ Medina y Galnares
16	17	1	2	50	2	3	C/ Medina y Galnares
16	18	1	2	50	2	2	Av. San Jerónimo
18	16	1	2	50	2	2	Av. San Jerónimo
17	18	1	1	40	2	2	C/ Tamarguillo
18	17	1	1	40	2	2	C/ Tamarguillo
18	19	1	2	50	2	2	C/ San Jerónimo
19	18	1	2	50	2	2	C/ San Jerónimo
7	19	1	2	50	2	3	Av. Biología
19	7	1	2	50	2	3	Av. Biología
19	7	1	2	50	2	3	Av. Biología
19	20	1	2	50	2	2	c/ Astronomía
20	19	1	2	50	2	2	c/ Astronomía
19	22	1	2	50	2	2	Av. Tecnología
22	19	1	2	50	2	2	Av. Tecnología
22	23	1	2	50	2	2	Av. Tecnología
23	22	1	2	50	2	2	Av. Tecnología
20	23	1	2	50	2	1	Av. Ingeniería
23	20	1	2	50	2	1	Av. ingeniería
20	21	1	1	40	2	1	C/ Estrella Betelguese
21	20	1	1	40	2	1	C/ Estrella Betelguese
21	9	1	1	40	2	1	C/ Estrella Canopus
9	21	1	1	40	2	1	C/ Estrella Canopus
20	6	1	1	40	2	1	Av. Ingeniería
6	20	1	1	40	2	1	Av. Ingeniería
23	24	1	2	50	2	2	C/ Parque Sierra de Castril
24	23	1	2	50	2	2	C/ Parque Sierra de Castril
24	25	1	2	50	2	3	Rda. Urbana norte
25	24	1	2	50	2	3	Rda. Urbana norte
21	24	1	1	40	2	2	C/ Estrella Canopus
24	21	1	1	40	2	2	C/ Estrella Canopus
25	10	1	1	40	2	3	C/ Agricultores
10	25	1	1	40	2	3	C/ Agricultores
25	26	1	2	50	2	3	Rda. Urbana norte
26	25	1	2	50	2	3	Rda. Urbana norte
26	27	1	1	40	2	1	C/ Cortijo de las casillas
27	26	1	1	40	2	1	C/ Cortijo de las

							casillas
26	37	1	2	50	2	3	Rda. Urbana norte
37	26	1	2	50	2	3	Rda. Urbana norte
10	27	800	3	80	2	2	Rda. Norte de Sevilla
27	10	800	3	80	2	2	Rda. Norte de Sevilla
27	28	1	3	80	2	2	Rda. Norte de Sevilla
28	27	1	3	80	2	2	Rda. Norte de Sevilla
28	29	1	3	80	2	2	Rda. Norte de Sevilla
29	28	1	3	80	2	2	Rda. Norte de Sevilla
29	30	1400	3	80	2	2	Rda. Norte de Sevilla
30	29	1400	3	80	2	2	Rda. Norte de Sevilla
30	31	750	3	80	2	2	Rda. Norte de Sevilla
31	30	750	3	80	2	2	Rda. Norte de Sevilla
31	32	1	3	80	2	2	Rda. Norte de Sevilla
32	31	1	3	80	2	2	Rda. Norte de Sevilla
33	31	1	1	40	2	2	C/ Sevilla Brenes
31	33	1	1	40	2	2	C/ Sevilla Brenes
33	34	1	3	80	2	3	Autovía del sur
34	33	1	3	80	2	4	Autovía del sur
34	35	1	2	50	2	2	Rda. Urbana norte
35	34	1	2	50	2	2	Rda. Urbana norte
35	36	1	2	50	2	3	Rda. Urbana norte
36	35	1	2	50	2	3	Rda. Urbana norte
36	37	1	2	50	2	3	Rda. Urbana norte
37	36	1	2	50	2	3	Rda. Urbana norte

La hoja Excel final contiene los 1266 arcos de la red perfectamente identificados.

Para el caso del análisis de la red basado en el flujo de personas se seleccionaron 164 zonas contenidas en las 6 macrozonas en las que se dividió la red (Macarena, Casco Histórico, Nervión, Sevilla Este, Triana y Zona Sur). Cada una de estas 164 contendrá un centroide, que son los puntos de donde salen y hacia donde se dirigen las personas. Estos centroides se unen a los nodos adyacentes del viario por medio de unos arcos especiales que se denominan conectores, representando el camino que tomarían los usuarios para incorporarse a la red y dirigirse a su destino. El número de personas que se desplazan de un centroide a otro se obtiene a partir de una matriz origen-destino, en adelante matrizOD.

6.2 Librería Networkx

Posteriormente a la recopilación de datos y dada la necesidad de calcular índices en un grafo extenso, se realizó un estudio de la librería Networkx de Python. La idea es utilizar en la medida de lo posible el soporte de las funciones de esta librería dado el elevado número de datos que definen el grafo, que a su vez provoca que el número de cálculos se dispare, por lo que es conveniente disponer de una herramienta de soporte en lenguaje de programación, en este caso Python.

La librería Networkx fue diseñada para trabajar con grafos y redes. Ofrece una amplia serie de herramientas y funciones que permiten crear, manipular y analizar estructuras de redes complejas. Estas pueden corresponder a grafos, dígrafos y multigrafos. Networkx además permite utilizar datos procedentes de Excel lo que ha resultado vital para el desarrollo del trabajo.

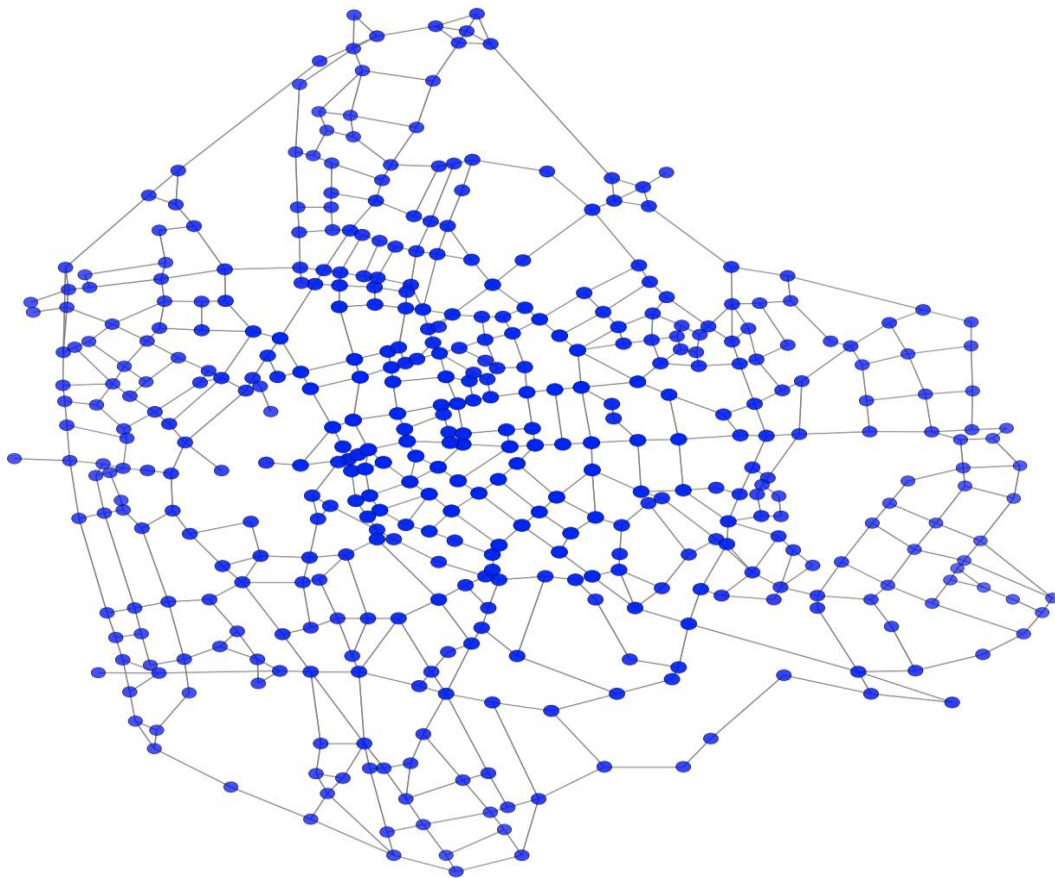
En esta búsqueda de funcionalidades a partir de artículos y búsqueda en la web se identificaron las funciones de Networkx que permiten calcular los índices Closeness, Betweenness y PageRank en el caso de redes topológicas y en algunos casos en redes con pesos.

6.3 Resultados

Una vez que se tienen recopilados todos los datos necesarios para el estudio se realizan los cálculos en Python con la librería Networkx y sus respectivas funciones. El uso de Python también ofrece la posibilidad de representar la red clasificando los nodos por valores como se muestra a continuación. Las ilustraciones de la 15 a la 20 se obtienen con el Programa 2.

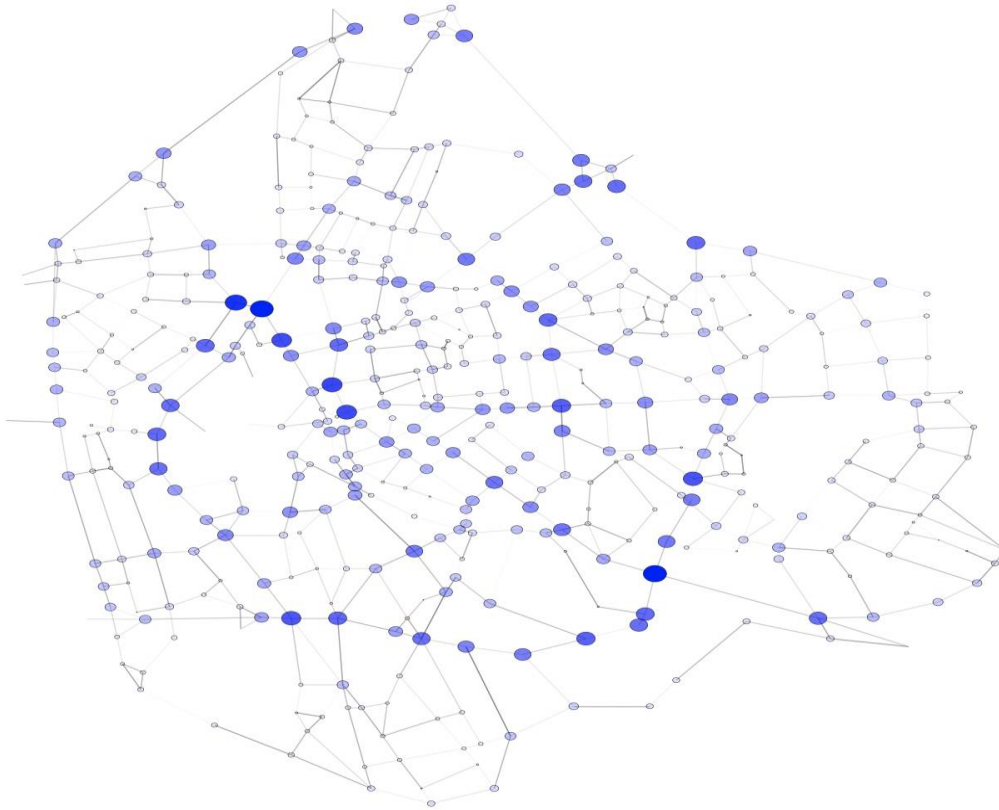
- Closeness Centrality puramente topológico.

Ilustración 15 Mapa con los nodos clasificados según su valor. Fuente: Elaboración propia.



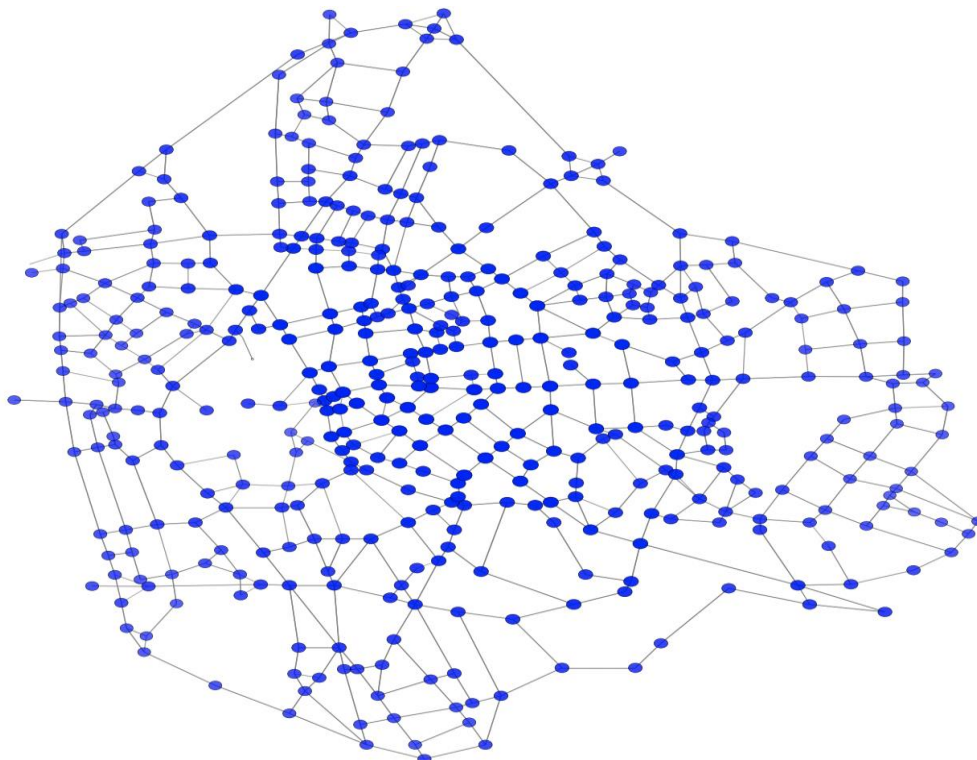
- Betweenness Centrality puramente topológico.

Ilustración 16 Mapa con los nodos clasificados según su valor. Fuente: Elaboración propia.



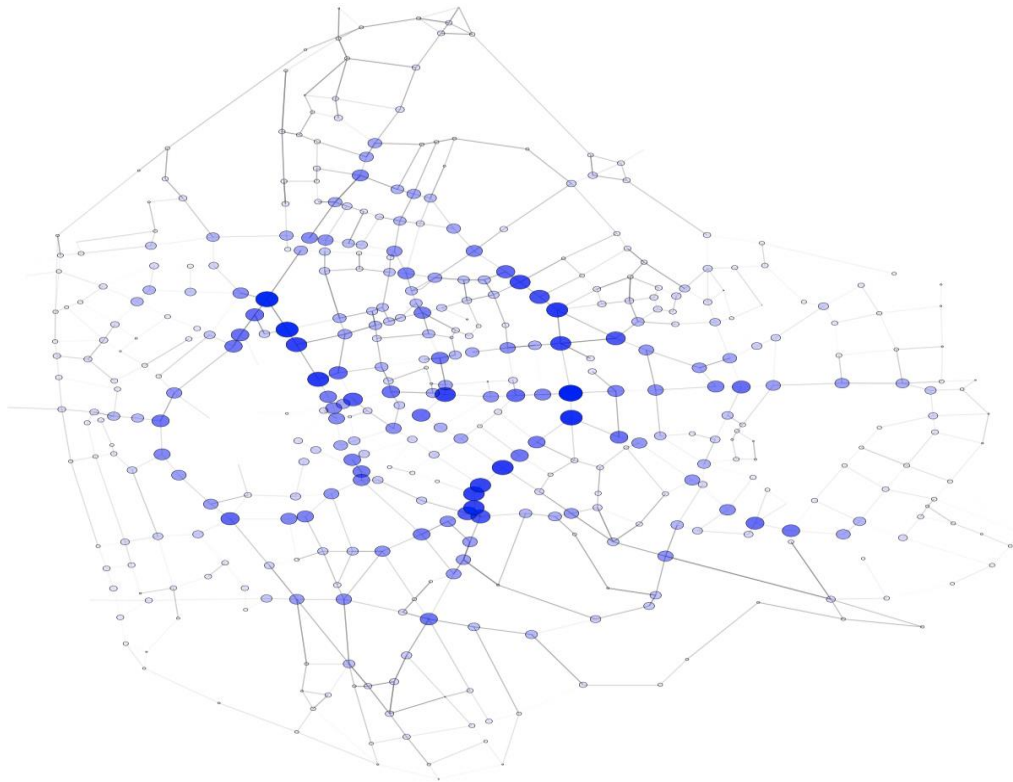
- Closeness Centrality.

Ilustración 17 Mapa con los nodos clasificados según su valor considerando atributos. Fuente: Elaboración propia.



- Betweenness Centrality.

Ilustración 18 Mapa con los nodos clasificados según su valor considerando atributos. Fuente: Elaboración propia.

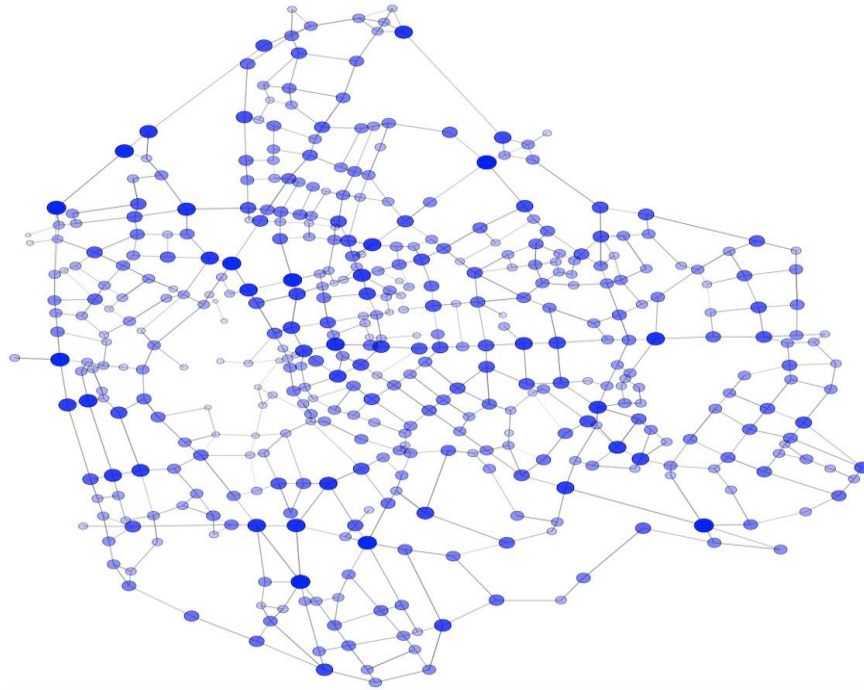


En las ilustraciones 15, 16, 17 y 18 se clasifican los nodos por su valor en Closeness Centrality y Betweenness Centrality respectivamente. Como se puede observar comparando las ilustraciones con las tablas de la 4 a la 9 de los rankings, se aprecia que los nodos están ordenados y que los de mayor valor son de mayor tamaño y de color más oscuro que los de menor valor. Para el caso de las ilustraciones sin atributos, se distinguen algunos nodos de gran tamaño en zonas no tan céntricas, por las cuales no deben pasar tantos caminos mínimos. Por ejemplo, se observa en la Ilustración 16 (Betweenness Centrality topológico) que los nodos de la Ronda Urbana Norte son más relevantes que en la Ilustración 18 (Betweenness Centrality).

En la siguiente Ilustración se muestra el mapa según los valores que resultan al aplicar el algoritmo de Pagerank. Aparecen nodos significativos por toda la red sin seguir un orden coherente, por lo que a la hora de comparar resultados y sacar conclusiones no se tienen en cuenta.

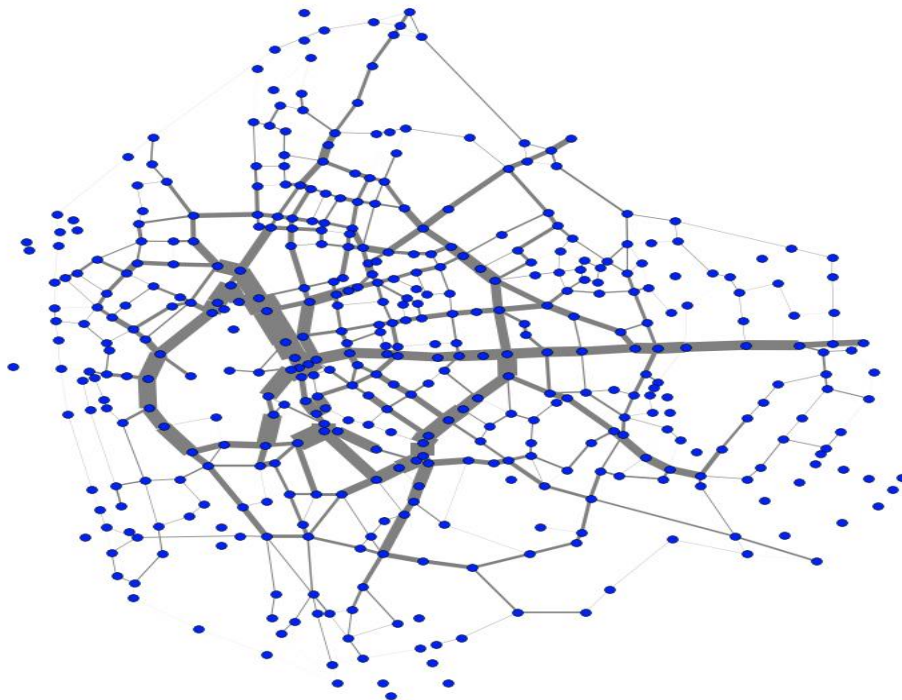
- Pagerank puramente topológico.

Ilustración 19 Mapa con los nodos clasificados según su valor. Fuente: Elaboración propia.



- Número de caminos mínimos por arco.

Ilustración 20 Número de caminos mínimos que pasan por cada arco. Fuente: Elaboración propia.

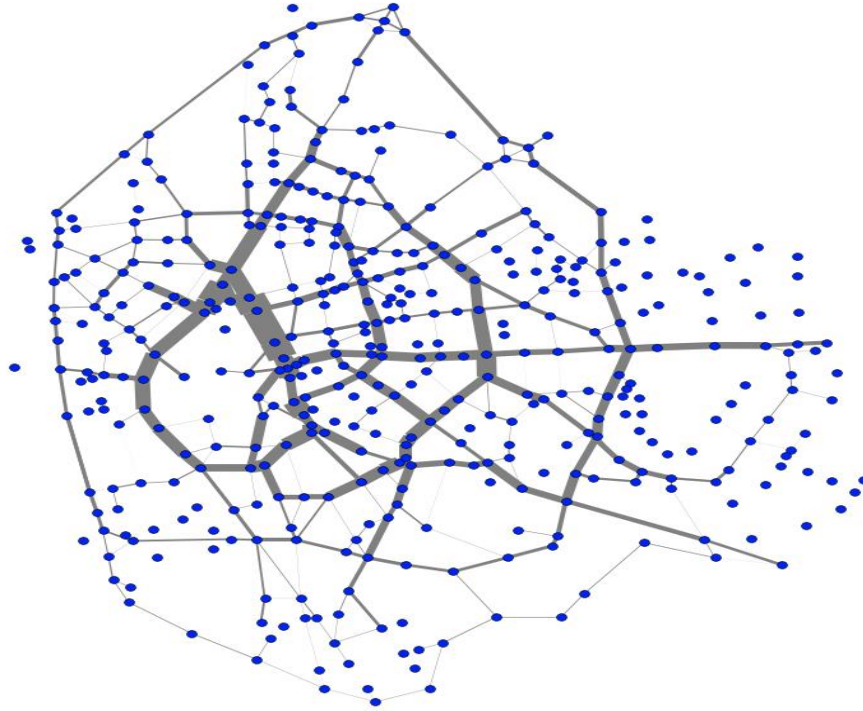


En la ilustración de caminos mínimos por arco se demuestra que por el viario principal, descrito en la Sección 5, es por donde se recogen más caminos mínimos. Para dicho cálculo, se tiene en

cuenta el tiempo, puesto que se saben las distancias de nodo a nodo y las velocidades de circulación por arco.

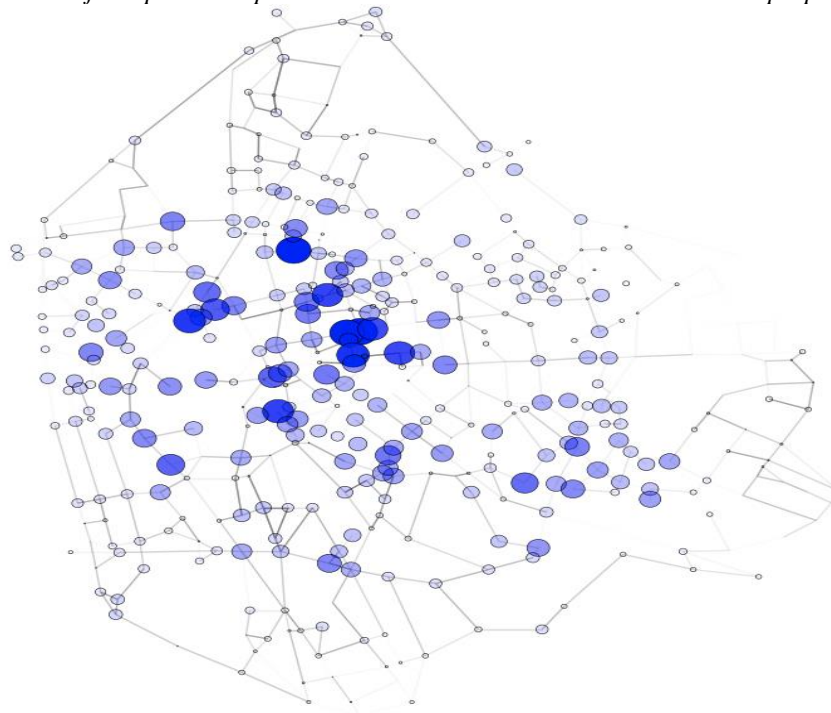
- Flujo de personas por arco.

Ilustración 21 Flujo de personas que circulan por cada arco. Fuente: Elaboración propia.



- Flujo de personas por nodo.

Ilustración 22 Flujo de personas que atraviesan cada nodo. Fuente: Elaboración propia.



Para el cálculo de flujo de personas por arco y por nodo se utiliza la demanda de la matrizOD para determinar el flujo de personas que entran y salen de cada centroide y que se incorporan a los nodos y arcos de la red a través de los conectores. Con estos datos podemos calcular el flujo de personas por arco y nodo mediante el código del Programa 1.

En las ilustraciones 21 y 22 se confirma que por los arcos y nodos que pasan la mayoría de los caminos mínimos es por donde más flujo de personas circulan, puesto que es el recorrido por el cual se tarda menos tiempo en llegar de un destino a otro.

6.4 Rankings

A partir del cálculo de índices fue posible obtener valores para cada nodo, para después normalizarlos y de esta forma realizar un ranking para cada uno de los índices, con el objetivo de compararlos y obtener conclusiones sobre los arcos y nodos más importantes de la red. En total se han realizado ocho rankings. Uno con cada índice, primero considerando medidas puramente topológicas, es decir, sin atributos y, posteriormente utilizando como pesos las distancias, el número de caminos mínimos que pasan por cada arco y el flujo de personas que atraviesan cada arco y cada nodo.

6.4.1 Comparación entre índices de Networkx

Una vez tenemos los resultados podemos ver cuales son los nodos más importantes según cada índice. Dado el elevado número de nodos y arcos, a fin de simplificar los resultados, en primer lugar, se comparan los rankings de Closeness y Betweenness usando los treinta primeros nodos según orden de mayor a menor valor de ambos índices. Se identifican aquellos nodos en los que hay coincidencia según los dos índices. Este análisis se repite para los nodos del 30 al 60 y finalmente con los nodos que ocupan las posiciones 70 al 100 en cada uno de los índices.

Así los nodos que resultan ser coincidentes en estos rankings son los siguientes:

- Treinta primeros nodos de mayor valor topológico

Tabla 4 Arcos formados por los nodos de más valor y su localización. Fuente: Elaboración propia.

BETWEENNES-CLOSENESS		
nodo 64		
64	65	Av. Kansas City
64	117	Av. Kansas City
64	110	C/ Efeso
64	63	Av. Alcalde Manuel del Valle
nodo 106		
106	105	Av. de Andalucía
106	107	Av. José María Javierre
106	190	Av. de Andalucía
106	195	C/ Clemente Hidalgo

nodo 134		
134	123	C/ Luis Montoto
134	133	C/ Luis Montoto
134	155	C/ Amador de los Ríos
134	156	C/ Juan Antonio Cavastany
134	173	C/ José María Moreno Galvan
nodo 195		
195	194	C/ Marqués de Pickman
195	196	C/ Navíos Argos
195	198	C/ Federico Mayo Agarre
195	220	Rda. Tamarguillo
195	106	C/ Clemente Hidalgo
nodo 173		
173	170	C/ Juan de Mata Carriazo
173	172	Av. de Eduardo Dato
173	152	C/ Demetrio de los rios
173	134	C/ Jose María Moreno Galván
nodo 220		
220	195	Av. Ronda Tamarguillo
220	198	Av. San Juan de la Cruz
220	219	Av. Ronda Tamarguillo
220	367	C/ Tarragona
220	368	C/ Paulo Orosio

- Del treinta al setenta de mayor valor topológico

Tabla 5 Arcos formados por los nodos de mayor valor y su localización. Fuente: Elaboración propia.

BETWEENNES-CLOSENESS		
nodo 198		
198	195	C/ Federia Mayo Agarre
198	199	C/ Amor
198	220	Av. San Juan de la Cruz
198	370	Av. Juan XXIII
Nodo 104		
104	103	Av. de Andalucía
104	98	C/ Comercio
104	105	Av. de Andalucía
104	199	Av. Parque Amate

nodo 157		
157	152	Av. Menéndez Pelayo
157	158	Av. el Cid
157	170	Av. Carlos V
nodo 218		
218	207	Av. Ramón y Cajal
218	217	Rda. de Tamarguillo
218	219	Rda. de Tamarguillo
218	369	Av. Hytasa
nodo 219		
219	220	Rda. de Tamarguillo
219	218	Rda. de Tamarguillo
219	364	C/ Afan de Rivera

- Del setenta al cien de mayor valor topológico

Tabla 6 Arcos formados por los nodos de mayor valor y su localización. Fuente: Elaboración propia.

BETWEENNES-CLOSENESS		
nodo 61		
61	59	Av. Alcalde Manuel del Valle
61	60	C/ Jose Pérez Ocaña
61	62	Av. Alcalde Manuel del Valle
nodo 217		
217	216	C/ Alberche
217	218	Av. Ronda de Tamarguillo
217	351	Av. Ronda de Tamarguillo
nodo 67		
67	34	Av. Kansas city
67	66	Av. Kansas city
67	68	C/ Macedonia
67	69	Av. de las Villas de Cuba
nodo 214		
214	212	Av. José Saramago
214	213	Av. José Saramago
214	215	Av. Alcalde Juan Fernández
214	221	C/ Nuestra Señora de las Mercedes

214	351	Av. de la Paz
nodo 138		
138	73	C/ María Auxiliadora
138	130	C/ María Auxiliadora
138	137	C/ Salesianos
nodo 60		
60	51	Av. de Llanes
60	59	Crta. de Carmona
60	61	C/ José Pérez Ocaña
60	71	Crta. de Carmona

Al analizar las tres tablas anteriores sobre los índices topológicos se consiguen unos buenos resultados experimentales, puesto que son los nodos que coinciden dentro del grupo de los cien primeros nodos con mayor valor de cada índice. Pero, por otro lado, no se han tenido cuenta las distancias reales entre nodos por lo que se consideran como una buena aproximación a la solución real.

- Treinta primeros nodos de mayor valor con distancias como atributos

Tabla 7 Arcos formado por los nodos de mayor valor y su localización. Fuente: Elaboración propia.

BETWEENNES-CLOSENESS		
nodo 115		
115	114	C/ Luis montoto
115	121	Av. Kansas City
115	186	C/ Luis Montoto
115	187	C/ Luis de Morales
nodo 123		
123	122	Av. de la Buhaira
123	134	c/ Luis Montoto
123	186	c/ Luis Montoto
123	174	Av. de la Buhaira
nodo 121		
121	115	Av. Kansas City
121	120	Av. Kansas City
121	122	C/ Pablo Picasso
121	124	Av. José Laguillo
nodo 134		

134	123	C/ Luis Montoto
134	133	C/ Luis Montoto
134	155	C/ Amador de los ríos
134	156	C/ Juan Antonio Cavastany
134	173	C/ José María Moreno Galvan
nodo 113		
113	112	Av. de el Greco
113	114	C/ Luis Montoto
113	189	Av. de la Cruz del Campo
113	190	C/ Luis Montoto
nodo 173		
173	170	C/ Juan de Mata Carriazo
173	172	Av. de Eduardo Dato
173	152	C/ Demetrio de los Ríos
173	134	C/ José María Moreno Galván
nodo 106		
106	105	Av. de Andalucía
106	107	Av. José María Javierre
106	190	Av. de Andalucía
106	195	C/ Clemente Hidalgo
nodo 64		
64	65	Av. Kansas city
64	117	Av. Kansas city
64	110	C/ Efeso
64	63	Av. Alcalde Manuel del Valle
nodo 195		
195	194	Marqués de Pickman
195	196	C/ Navío Argos
195	198	C/ Federico Mayo Agarre
195	220	Av. Ronda tamarguillo
195	106	C/ Clemente Hidalgo

- Del treinta al setenta de mayor valor con distancias como atributos

Tabla 8 Arcos formados por los nodos de mayor valor y su localización. Fuente: Elaboración propia.

BETWEENNES-CLOSENESS		
nodo 135		
135	132	Av. Menéndez Pelayo

135	133	C/ La forida
135	152	Av. Menéndez Pelayo
nodo 190		
190	106	Av. de Andalucía
190	113	C/ Luis Montoto
190	194	C/ Mariano Benlliure
nodo 210		
210	180	Av. Ramón y Cajal
210	181	Av. San Francisco Javier
210	203	C/ Aspinosa y Cárcel
210	209	Av. Ramón y Cajal
210	211	Av. San Francisco Javier
nodo 131		
131	130	C/ Recadero
131	132	C/ Recadero
131	154	C/ Guadalupe
131	155	C/ Juan de Vera
nodo 110		
110	64	C/ Efeso
110	109	C/ Dr Laffon Soto
110	111	C/ Efeso
110	116	Av. de la Solea
nodo 111		
111	107	C/ de la Ada
111	108	C/ Jerusalén
111	110	C/ Efeso
111	112	C/ Tarso

- Del setenta al cien de mayor valor con distancias como atributos

Tabla 9 Arcos formados por los nodos de mayor valor y su localización. Fuente: Elaboración propia.

BETWEENNES-CLOSENESS		
nodo 214		
214	212	Av. José Saramago
214	213	Av. José Saramago
214	215	Av. Alcalde Juan Fernández
214	221	C/ Nuestra Señora de las Mercedes
214	351	Av. de la Paz

Las tablas 7, 8 y 9 muestran las avenidas y calles formadas por los nodos de mayor valor en ambos índices teniendo en cuenta distancias. Por lo tanto, se pueden obtener las intersecciones más vulnerables de la red, la intersección entre la avenida Kansas city y la calle Efeso, entre las calles Luis Montoto, Recadero y la avenida Menéndez Pelayo, entre la calle Luis Montoto, y las avenidas Cruz del campo y el Greco, entre las avenidas Andalucía y Clemente Hidalgo y entre las avenidas Eduardo Dato y Menéndez Pelayo son algunos ejemplos entre otros.

6.4.2 Número de caminos mínimos por arco

Para el cálculo de los caminos mínimos entre nodos se utiliza como variable el tiempo, puesto que se sabe las distancias de nodo a nodo y las velocidades de circulación por arco. Con estos datos y la función de Networkx `nx.shortest_path` se obtiene que los 50 arcos más relevantes de la red son los siguientes.

Tabla 10 Ranking de los 50 arcos por los cuales pasan más caminos mínimos. Fuente: Elaboración propia.

Arcos	Localización
(157, 158)	Av. el Cid
(135, 152)	Av. Menéndez Pelayo
(152, 157)	Av Menéndez Pelayo
(160, 162)	Paseo de las Delcias
(162, 164)	Paseo de las Delicias
(131, 132)	C/ Recadero
(158, 159)	Av. de María Luisa
(73, 138)	C/ María Auxiliadora
(130, 138)	C/ María Auxiliadora
(132, 135)	Av. Menéndez Pelayo
(102, 404)	Av. de Andalucía
(106, 107)	Av. José María Javierre
(51, 60)	Av. de Llanes
(402, 403)	A-92
(403, 404)	A-92
(106, 190)	Av. de Andalucía
(34, 84)	Autovía del sur
(101, 102)	SE-30, 41020 Sevilla
(159, 235)	Puente de los Remedios
(148, 164)	Paseo Cristóbal Colón
(113, 114)	C/ Luis Montoto
(102, 103)	Av. de Andalucía
(113, 190)	C/ Luis Montoto
(100, 101)	SE-30, 41020 Sevilla
(83, 97)	Autovía del sur

(147, 148)	C/ Arjona
(159, 229)	Paseo de las Delicias
(114, 115)	C/Luis Montoto
(123, 134)	C/ Luis Montoto
(133, 134)	C/ Luis Montoto
(97, 100)	Autovía del sur
(160, 161)	C/ de la Rabida
(51, 52)	Av. de Llanes
(59, 60)	Ctra. Carmona
(146, 147)	C/ Torneo
(62, 63)	Av. Alcalde Manuel del Valle
(63, 64)	Av. Alcalde Manuel del Valle
(219, 220)	Av. Ronda del tamarguillo
(143, 144)	C/ Torneo
(158, 161)	C/ Palos de la Frontera
(378, 388)	Av. del Sur
(102, 388)	Autovía del sur
(64, 110)	C/ Efeso
(61, 62)	Av. Alcalde Manuel del Valle
(130, 131)	C/ Recadero
(133, 135)	C/ la Florida
(221, 222)	Cardenal Buenomonreal
(83, 84)	Autovía del sur

6.4.3 Flujo de personas que circulan por la red

Una vez se tiene el flujo de personas que circula por cada arco y nodo, se hace un ranking de la siguiente forma: se divide el valor de cada arco o nodo individual entre el flujo total de personas en la red y después se procede a normalizar estos valores para facilitar su clasificación. En el caso de los arcos, se sumó el flujo del nodo origen más el flujo del nodo destino y seguidamente se siguió el proceso anterior.

Tabla 11 Ranking de los 50 arcos con mayor flujo de personas y su localización. Fuente: Elaboración propia.

Arco	Localización
(157, 158)	Av. el Cid
(152, 157)	Av. Menéndez Pelayo
(135, 152)	Av. Menéndez Pelayo
(160, 162)	Paseo de las Delcias
(162, 164)	Paseo de las Delicias
(131, 132)	C/ Recadero
(106, 107)	Av. José María Javierre
(132, 135)	Av. Menéndez Pelayo
(158, 159)	Av. de María Luisa

(73, 138)	C/ María Auxiliadora
(130, 138)	C/ María Auxiliadora
(159, 229)	Paseo de las Delicias
(160, 161)	C/ de la Rábida
(219, 220)	Av. Ronda del Tamarguillo
(147, 148)	C/ Arjona
(159, 235)	Puente de los Remedios
(158, 161)	C/ Palos de la Frontera
(148, 164)	Paseo de Cristobal Colón
(146, 147)	C/ Torneo
(164, 244)	Puente de San Telmo
(123, 134)	C/ Luis Montoto
(133, 134)	C/ Luis Montoto
(218, 219)	Av. Ronda Tamarguillo
(62, 63)	Av. Alcalde Manuel del Valle
(63, 64)	Av. Alcalde Manuel del Valle
(106, 195)	C/ Clemente Hidalgo
(106, 190)	Av. de Andalucía
(229, 230)	Av. de la palmera
(243, 244)	Av. República Argentina
(113, 190)	C/ Luis Montoto
(113, 114)	C/ Luis Montoto
(143, 144)	C/ Torneo
(115, 187)	C/ Luis de Morales
(217, 218)	Av. Ronda del Tamarguillo
(51, 60)	Av. de Llanes
(130, 131)	C/ Recadero
(114, 115)	C/ Luis Montoto
(64, 110)	C/ Efeso
(117, 120)	Av. Kansas city
(76, 77)	C/ Resolana
(221, 222)	Cardenal Buenomonreal
(61, 62)	Av. Alcalde Manuel del Valle
(64, 117)	Av. Kansas city
(110, 111)	C/ Efeso
(230, 344)	Av. de la palmera
(107, 111)	C/ de la Ada
(120, 121)	Av. Kansas city
(144, 146)	C/ Torneo

Tabla 12 Ranking de los 50 nodos con mayor flujo de personas. Fuente: Elaboración propia.

Nodo	Flujo
157	221683
152	220640
158	214120
159	191482
135	184358
106	175052
164	171932
160	168174
162	163388
132	147922
115	138933
107	136598
64	131669
131	131577
60	120032
102	118666
148	114581
147	114414
73	111372
218	109820
130	107448
230	107190
59	105281
229	105031
138	103788
121	103627
195	101508
210	101349
220	100467
161	98317
123	98260
219	96471
182	93570
34	93222
244	91244
344	89952
134	88162
133	88162
63	87162
113	84512
190	83873

77	82738
146	82682
76	82284
62	80997
338	79464
78	78822
114	78628

Comparando los resultados mostrados en las tablas 11 y 12, se observa que coinciden los nodos de mayor flujo con los nodos que forman los arcos de mayor flujo. Se muestran los 50 arcos y nodos mas relevantes de la red. Se vuelven a obtener las calles y avenidas que forman las intersecciones más vulnerables de la red, además del puente de los Remedios y San Telmo que hacen posible la conexión del flujo debido a la situación geográfica de la ciudad.

7 CONCLUSIONES

Tran analizar los resultados se obtienen los nodos que actúan como puntos críticos de control de flujo a través de la red de transporte. Los nodos con un alto valor en los índices son esenciales para la conectividad y eficiencia de la red, pero también representan los puntos vulnerables. Un ataque o fallo en estos nodos podría tener consecuencias significativas en términos de interrupción del servicio y disminución de la eficiencia del sistema.

En este trabajo se han tenido en cuenta las distancias, el tiempo de desplazamiento entre nodos, así como el flujo de personas que circulan por la red. La integración de esta última perspectiva ha aportado una comprensión más profunda de la vulnerabilidad de la red y sus implicaciones prácticas. Mediante la elaboración de un código en un lenguaje de programación de vigencia actual, tomando datos reales y utilizando fundamentos matemáticos se obtienen resultados con posibles futuras aplicaciones.

Se demuestra que los componentes más vulnerables de la red de transporte de Sevilla están en las rondas exteriores, la ronda metropolitana SE-30 formada por el puente del Centenario y por el puente del Rey Juan Carlos I, desempeñando un papel crucial para el acceso de entrada y salida de la ciudad.

Las tres vías más relevantes dentro del viario principal son la calle Luis Montoto, la avenida de María Luisa y la avenida de Andalucía. Articulan la movilidad de la red puesto que por ellas pasan la mayoría de los caminos mínimos y son el pulso de la ciudad al recaer en ellas el mayor flujo de personas.

A su vez, en el viario secundario cabe resaltar el papel de las vías Méndez Pelayo, Recadero, el Cid, Torneo, paseo de las Delicias y Efeso. Su elevado volumen de tráfico, en algunos casos superior al de las vías principales, se debe a que marcan la interconexión entre los diferentes barrios de la ciudad.

Por último, nombrar los puentes urbanos de San Telmo, las Delicias y de los Remedios, que son esenciales debido a la situación geográfica de la ciudad de Sevilla y por los cuales se distribuyen un número elevado del flujo de la red.

La realización de este proyecto me ha permitido enfrentarme a la situación de resolver una problemática real como es el estudio de una red de transporte, con datos en su gran mayoría reales y haciendo uso de herramientas precisas y sofisticadas. Situaciones tales como analizar un elevado número de resultados me han permitido valorar la importancia de decidir una estrategia a seguir en la resolución de un problema. También me ha permitido comprender como afecta la geografía de una ciudad en el análisis de la vulnerabilidad de una red de transporte.

Para terminar, señalar una posible continuación del trabajo que consistiría en eliminar nodos y enlaces importantes de la red y analizar como afectarían estas acciones al número de caminos mínimos, y analizar la forma en que se distribuye nuevamente el flujo de personas, obteniendo nuevas medidas que ayudarían a mejorar la clasificación generada en este estudio.

BIBLIOGRAFÍA

- Allenby, B., & Fink, J. (2005). Toward Inherently Secure and Resilient Societies. *science*, 309 (5737), 1034-1036.
- Appert, M., & Chapelon, L. (2013). Measuring urban road network vulnerability using graphtheory : the case of Montpellier's road network. *HAL science ouverte*, 1.
- Barrat, A., Bathelemy, M., & Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101 (11), 3747-3752.
- Berdica, K. (2002). An introduction to road vulnerability: what has been done, is done and should be done. *Transport Policy*, 9 (2), 117-227.
- Boccaletti, G., Latora, V., & Chavez, M. (2006). Complex networks: Structure and dynamics. *Physics Reports*, 424 (4-5), 175-308.
- Boldi, P., & Vigna, S. (2013). Axioms for Centrality. *Internet Mathematics*, 10, 222-262.
- Borgato Morelli, A., & Luiz Cunha, A. (2021). Measuring urban road network vulnerability to extreme events: An application for urban floods. *Transportation Research Part D: Transport and Environment*, 93, 1361-9209.
- Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30 (1-7), 107-117.
- Cats, O., & Jenelius, E. (2015). Planning for the unexpected: The value of reserve capacity for public transport network robustness. *Transportation Research Part A: Policy and Practice*, 81, 47-61.
- Cox, A., & Rose, A. (2011). Transportation security and the role of resilience: A foundation for operational metrics. *Transport Policy*, 18 (2), 307-317.
- Dong, X., Guan, J., & Xing, L. (2019). Betweenness centrality for similarity-weight network and its application to measuring industrial sectors' pivotability on the global value chain. *Statistical Mechanics and its Applications* , 516, 19-36.
- Feng, X., & Levinson, D. (2011). Evaluating the effects of the I-35W bridge collapse on road-users in the twin cities metropolitan region. *Transportation Planning and Technology* , 34 (7), 691-703.
- Freeman, L. (1977). A Set of Measures of Centrality Based on Betweenness. *American Sociological Association*, 40 (1), 35-41.
- Gallagher, D. (2015). Planning under new extremes: Resilience and the most vulnerable. *ICE Proceedings Municipal Engineer*, 169 (3), 1-12.
- Gedik, R., Medal, H., & Rainwater, C. (2014). Vulnerability assessment and re-routing of freight trains under disruptions: A coal supply chain network application. *Transportation Research Part E: Logistics and Transportation Review*, 71, 45-57.
- Holling, C. (1973). Resilience and Stability of Ecological Systems. *Annual Review of Ecology and Systematics*, 4, 1-23.
- Huang, T., & Zhang, M. (2022). Complex-network-based traffic network analysis and dynamics: A comprehensive review. *Physica A: Statistical Mechanics and its Applications*, 607.
- IMD. (2016). Obtenido de <https://www.sevilla.org/actualidad/blog/plan-de-movilidad-urbana-sostenible-de-sevilla>
- IMD. (2016). Obtenido de <https://www.sevilla.org/actualidad/blog/plan-de-movilidad-urbana-sostenible-de-sevilla>
- Jie, L., & Zhenwu, S. (2021). Measuring the dynamic evolution of road network vulnerability to floods: A case study of Wuhan, China. *Travel Behaviour and Society*, 23, 13-24.
- Kaplan, S., & Garrick, B. (1981). Obtenido de <https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1539-6924.1981.tb01350.x>

- Kun, J., Wei, W., & Xinram, L. (2022). Identifying the critical road combination in urban roads network under multiple disruption scenarios. *Physica A: Statistical Mechanics and its Applications*, 607, 378-4371.
- Lars-Göran, M., & Jenelius, E. (2015). Vulnerability and resilience of transport systems – A discussion of recent research. *Transportation Research Part A: Policy and Practice*, 81, 16-34.
- Leal De Oliveira, E., & Da Silva Portugal, L. (2014). Determining Critical Links in a Road Network: Vulnerability and Congestion Indicators. *Procedia - Social and Behavioral Sciences*, 162, 158-167.
- Latora, V. (2001). Efficient Behavior of Small-World Networks. *Physical Review Letters*, 87 (19), 4-13.
- M'cleod, L., Vecsler, R., & Yuan, S. (2017). Vulnerability of Transportation Networks: The New York City Subway System under Simultaneous Disruptive Events. *Procedia Computer Science*, 119, 42-50.
- Ma, J., & Wang, P. (2023). The influence of layered community network structure on traffic capacity. *Physica A: Statistical Mechanics and its Applications*, 626.
- Manikan, P. (2017). Resilience Engineering: An Integrative Review of Fundamental Concepts and Directions for Future Research in Safety Management. *Open Journal of Safety Science and Technology*, 7 (4), 129-160.
- Mattsson, L.-G., & Jenelius, E. (2015). Vulnerability and resilience of transport systems – A discussion of recent research. *Transportation Research*, Part A 81, 16–34.
- Newman, M. (2003). The Structure and Function of Complex Networks. *SIAM*, 45 (2), 167-256.
- Nicholson, P. (2007). Road network unreliability: impact assessment and mitigation. *International Journal of Critical Infrastructures*, 3 (3-4), 346-375.
- Olsson, L. (2005). Road investment scenarios in Northern Sweden. *Forest Policy and Economics*, 7 (4), 615-623.
- Rodríguez Nuñez, E., & García Palomares, J. (2014). Measuring the vulnerability of public transport networks. *Journal of Transport Geography*, 35, 50-63.
- Sun, D., & Zhao, Y. (2015). Vulnerability Analysis of Urban Rail Transit Networks: A Case Study of Shanghai, China. *Sustainability*, 7 (6), 6919-6936.
- Sunghoon, K., & Hwasoo, Y. (2016). A Flow-based Vulnerability Measure for the Resilience of Urban Road Network. *Procedia - Social and Behavioral Sciences*, 218, 13-23.
- Taylor, M. (2003). *The Network Reliability of Transport*.
- Wang, M., & Zhang, J. (2019). Transportation functionality vulnerability of urban rail transit networks based on movingblock: The case of Nanjing metro. *Physica A: Statistical Mechanics and its Applications*, 535.
- Wang, Z., Pei, Y., & Liu, J. (2023). Vulnerability analysis of urban road networks based on traffic situation. *International Journal of Critical Infrastructure Protection*, 41.
- Wei, Y., Wu, J., & Yin, N. (2023). An integrated resilience assessment model of urban transportation network: A case study of 40 cities in China. *Transportation Research Part A: Policy and Practice*, 173.
- Wu, C. (2005). Synchronizability of networks of chaotic systems coupled via a graph with a prescribed degree sequence. *Physics Letters A*, 436 (4), 281-287.
- Wu, C.-Y., & Hu, M.-B. (2021). Effects of road network structure on the performance of urban traffic systems. *Physica A: Statistical Mechanics and its Applications*, 563.
- Yingying, D., & Feng, L. (2013). Structural robustness of city road networks based on community. *Computers, Environment and Urban Systems*, 41, 75-87.
- Zanjirani Farahan, R., & Miandoabchi, E. (2013). A review of urban transportation network design problems. *European Journal of Operational Research*, 229 (2), 281-302.
- Zarghami, S., & Dumark, J. (2021). Unearthing vulnerability of supply provision in logistics networks to the black swan events: Applications of entropy theory and network analysis. *Reliability Engineering & System Safety*, 215.

ANEXO. CÓDIGO DE PYTHON

Programa 1. Resolución del cálculo de los índices.

```
# See PyCharm help at https://www.jetbrains.com/help/pycharm/

from IOfunctionsExcel5 import *
import math
import networkx as nx
import pandas as pd
from matplotlib import pyplot as plt

Book='Red sevilla ampliada.xlsx'
Hoja='Hoja1'

Rangos={
'Nodos':['A2','C436'],
'Arcos':['H2','O718'],
'EscalaN1':['W4'],
'EscalaN2':['W6'],
'Escala':['W8'],

}

Nodos=Read_Excel_to_NesteDic(Book,Hoja, Rangos['Nodos'][0],
Rangos['Nodos'][1])
Arcos=Read_Excel_to_NesteDic_TupleInt(Book,Hoja, Rangos['Arcos'][0],
Rangos['Arcos'][1])
Nodo1=Read_Excel_Cell(Book,Hoja, Rangos['EscalaN1'][0])
Nodo2=Read_Excel_Cell(Book,Hoja, Rangos['EscalaN2'][0])
DistReal_N1N2=Read_Excel_Cell(Book,Hoja, Rangos['Escala'][0])
Distancia_N1_N2=math.sqrt((Nodos[Nodo1]['X']-
Nodos[Nodo2]['X'])**2+(Nodos[Nodo1]['Y']-Nodos[Nodo2]['Y'])**2 )

Factor_Escala= DistReal_N1N2/Distancia_N1_N2

#Si quisieramos duplicar arcos con doble sentido
Duplicados={}
for (i,j) in Arcos:
    if Arcos[(i,j)]['Sentidos']==2:
        Duplicados.update({(j,i):{'Origen':j, 'Destino':i,
'Factor':Arcos[(i,j)]['Factor'],
'Tipo':Arcos[(i,j)]['Tipo'],'Velocidad':Arcos[(i,j)]['Velocidad'],
'Sentidos':Arcos[(i,j)]['Sentidos'],'carriles':Arcos[(i,j)]['carriles'] }})

Arcos.update(Duplicados)
```

```

# Crear Longitudes calculadas de todos Los arcos

Longitudes={}
tiempo={}#para La segunda parte
for (i,j) in Arcos:

    if Arcos[(i, j)]['Factor'] == 1:
        Longitudes[(i, j)] = Factor_Escala * math.sqrt((Nodos[i]['X'] -
Nodos[j]['X']) ** 2 + (Nodos[i]['Y'] - Nodos[j]['Y']) ** 2)
        tiempo[(i,j)]=60*(((Longitudes[(i,
j))]/1000)/Arcos[(i,j)]['Velocidad'])#para La segunda parte
    else:
        Longitudes[(i, j)] = Arcos[(i, j)]['Factor'];
        tiempo[(i, j)] = 60*(((Longitudes[(i, j)])/1000) / Arcos[(i,
j)]['Velocidad'])#para La segunda parte

Tipo_velocidad={}

for (i,j) in Arcos:
    Tipo_velocidad[(i,j)]= Arcos[(i,j)]['Velocidad'] #en verde hayque ponerel
nombre que venga en La columna del excel

Tipo_carriles={}

for(i,j) in Arcos:
    Tipo_carriles[(i,j)]= Arcos[(i,j)]['carriles']

G=nx.DiGraph()
ListaNodos=[i for i in Nodos]
ListaArcos=[(i,j,Longitudes[i,j]) for (i,j) in Arcos] #poner tiempo para La
segunda parte o Longitud para La primera
G.add_nodes_from(ListaNodos)
G.add_weighted_edges_from(ListaArcos)
NodosPosiciones={i:[Nodos[i]['X'],Nodos[i]['Y']] for i in Nodos}

nx.set_edge_attributes(G, Longitudes, name='Long')
nx.set_edge_attributes(G, Tipo_velocidad, name='Velocidad')
nx.set_edge_attributes(G, Tipo_carriles, name='carriles')
nx.set_edge_attributes(G, Tipo_velocidad, 'etiqueta')#para poder poner
etiquetas a Los arcos
#para poder poner etiquetas a Los arcos
for u, v, data in G.edges(data=True):
    etiqueta = data['etiqueta']
    #print(f'Arco ({u}, {v}) - Etiqueta: {etiqueta}')

#funcion elimina arcos
def eliminar_arcos(grafo, arc_list):
    for arco in arc_list:
        if grafo.has_edge(*arco):
            grafo.remove_edge(*arco)

#funcion elimina nodos
def eliminar_nodos(grafo, nodo_list):

```

```

    grafo.remove_nodos_from(nodo_list)

#nodos_a_eliminar = [10, 300]
#arcos_a_eliminar = [(27, 28), (28, 29), (28,27), (29,28)]
#grafo_copia = G.copy()
# Eliminar los nodos o arcos
#eliminar_arcos(grafo_copia, arcos_a_eliminar )
#print( nx.closeness_centrality(grafo_copia))#medidas topológicas
nx.draw_networkx(G,NodosPosiciones, with_labels=True, node_size=9,
font_size=6, arrowstyle='-')
#etiquetas_de_arcos = nx.get_edge_attributes(G, 'etiqueta')
#nx.draw_networkx_edge_labels(G, NodosPosiciones, font_color='red',
font_size= 5)

plt.title('Grafo Resultante')
plt.show()

for node in G.nodes():
    peso1=0
    outedges = list(G.out_edges(node)) #proporciona el numero de arcos que
salen de un nodo
    inedges = list(G.in_edges(node)) #nos dice el numero de arcos que
entran en un nodo
    print(f"Node {node}:")
    print(f"\tCarriles que llegan: {inedges}")
    print(f"\tCarriles que salen: {outedges}")
    print()
    for (i,j) in Arcos:
        if node == Arcos[(i, j)]['Destino'] and Arcos[(i,j)]['Sentidos']==2:
            peso1 =peso1+ Arcos[(i, j)]['carriles']*2
        else:
            if (node == Arcos[(i, j)]['Destino'] or node == Arcos[(i,
j)]['Origen']) and Arcos[(i,j)]['Sentidos']==1:
                peso1 = peso1 + Arcos[(i, j)]['carriles']
    print(peso1)

#para ver a que nodos no les llega ningun arco, solo salen de él
def nodo_sin_fuente(grafo, nodo):
    for n in G.nodes():
        if n != nodo and grafo.has_edge(n, nodo):
            return False
    return True

for nodo in G.nodes():
    if nodo_sin_fuente(G, nodo):
        print(nodo)

# Calcular el PageRank
pagerank = nx.pagerank(G, alpha=0.85, max_iter=100, tol=1e-06)
ranking = sorted(pagerank.items(), key=lambda x: x[1], reverse=True)
# Imprimir el ranking

```

```

for i, (nodo, centrality) in enumerate(ranking):
    print(f"Puesto {i+1}: Nodo {nodo}, Pagerank = {centrality}")
# Imprimo Los valores de PageRank para cada nodo
for node, score in pagerank.items():
    print(f"Nodo {node}: PageRank = {score}")

#ver la centralidad de un nodo
# Calcular la centralidad de cercanía para todos los nodos
for node in G.nodes():
    closeness centrality = nx.closeness centrality(G, distance='Long')
#distance= Longitudes[(i,j)], sale igual dándole valor 1 a cada distancia

betweenness=nx.betweenness centrality(G, weight='Long')#medida con distancias
ranking2 = sorted(betweenness.items(), key=lambda x: x[1], reverse=True)
# Imprimir el ranking
for i, (nodo, centrality) in enumerate(ranking2):
    print(f"Puesto {i+1}: Nodo {nodo}, Betweenness centrality= {centrality}")

betweenness_topo=nx.betweenness centrality(G)#medidas topológicas
ranking3 = sorted(betweenness_topo.items(), key=lambda x: x[1], reverse=True)
# Imprimir el ranking
for i, (nodo, centrality) in enumerate(ranking3):
    print(f"Puesto {i+1}: Nodo {nodo}, Betweenness centrality topo=
{centrality}")

closeness= nx.closeness centrality(G, distance='Long')#medida con distancias
ranking4 = sorted(closeness.items(), key=lambda x: x[1], reverse=True)
# Imprimir el ranking
for i, (nodo, centrality) in enumerate(ranking4):
    print(f"Puesto {i+1}: Nodo {nodo}, Closeness centrality= {centrality}")

closeness_topo= nx.closeness centrality(G) #medida topologica
ranking5 = sorted(closeness_topo.items(), key=lambda x: x[1], reverse=True)
# Imprimir el ranking
for i, (nodo, centrality) in enumerate(ranking5):
    print(f"Puesto {i+1}: Nodo {nodo}, Closeness centrality topo=
{centrality}")
plt.axis('equal')
nx.draw_networkx(G,NodosPosiciones, with_labels=True, node_size=5,
font_size=6, arrowstyle='-') #with_labels=True,
plt.show()
#print(a)

#segundo parte, estudio con flujo de personas y tiempo entre arcos
Listaconectores=[(1001,77,5),(1002,143,5),(1002,145,5),(1002,142,5),(1003,141
,5),(1004,75,5),(1005,144,5),(1005,146,5),(1006,153,5),(1007,139,5),(1007,153
,5),(1007,130,5),(1008,144,5),(1008,146,5),(1009,147,5),
(1010,149,5),(1010,153,5),(1011,154,5),(1012,165,5),(1012,164,5),(1013,149,5)
,(1014,164,5),(1015,145,5),(1015,140,5),(1016,343,5),(1016,340,5),(1017,263,5)

```

), (1017, 264, 5), (1018, 257, 5), (1019, 257, 5), (1019, 255, 5), (1020, 252, 5), (1020, 248, 5),
 (1021, 240, 5), (1021, 266, 5), (1022, 246, 5), (1023, 246, 5), (1023, 248, 5), (1024, 242, 5),
 (1025, 243, 5), (1026, 250, 5), (1027, 264, 5), (1027, 265, 5), (1028, 241, 5), (1028, 242, 5),
 (1029, 235, 5), (1029, 236, 5), (1030, 235, 5), (1030, 236, 5),
 (1031, 234, 5), (1031, 237, 5), (1032, 234, 5), (1033, 301, 5), (1033, 303, 5), (1034, 305, 5),
 (1034, 306, 5), (1035, 233, 5), (1036, 308, 5), (1036, 309, 5), (1037, 286, 5), (1038, 237, 5),
 (1039, 278, 5), (1039, 292, 5), (1040, 271, 5),
 (1041, 148, 5), (1042, 158, 5), (1042, 159, 5), (1043, 159, 5), (1044, 169, 5), (1044, 170, 5),
 (1045, 222, 5), (1046, 157, 5), (1046, 161, 5), (1047, 349, 5), (1047, 350, 5), (1048, 229, 5),
 (1048, 231, 5), (1049, 322, 5), (1049, 323, 5), (1050, 342, 5), (1050, 343, 5),
 (1051, 159, 5), (1051, 232, 5), (1052, 231, 5), (1052, 232, 5), (1053, 347, 5), (1053, 346, 5),
 (1053, 347, 5), (1054, 336, 5), (1055, 331, 5), (1055, 337, 5), (1056, 318, 5), (1056, 320, 5),
 (1056, 322, 5), (1057, 352, 5), (1057, 353, 5), (1057, 354, 5), (1058, 251, 5), (1059, 353, 5),
 (1059, 354, 5),
 (1060, 149, 5), (1060, 150, 5), (1061, 353, 5), (1061, 354, 5), (1062, 329, 5), (1063, 315, 5),
 (1063, 330, 5), (1064, 359, 5), (1065, 317, 5), (1065, 341, 5), (1066, 329, 5), (1067, 129, 5),
 (1067, 131, 5), (1068, 186, 5), (1068, 121, 5), (1069, 39, 5), (1069, 65, 5), (1069, 66, 5),
 (1070, 114, 5), (1070, 115, 5), (1070, 120, 5), (1070, 121, 5), (1071, 152, 5), (1072, 172, 5),
 (1072, 175, 5), (1073, 167, 5), (1073, 223, 5), (1074, 187, 5), (1074, 191, 5), (1074, 192, 5),
 (1075, 189, 5), (1075, 190, 5), (1075, 194, 5), (1076, 193, 5), (1076, 202, 5), (1076, 206, 5),
 (1077, 417, 5),
 (1078, 194, 5), (1078, 206, 5), (1079, 152, 5), (1079, 157, 5), (1080, 180, 5), (1080, 210, 5),
 (1081, 181, 5), (1081, 183, 5), (1082, 207, 5), (1082, 219, 5), (1083, 210, 5), (1083, 214, 5),
 (1084, 123, 5), (1084, 174, 5), (1085, 214, 5), (1085, 216, 5), (1085, 351, 5), (1086, 207, 5),
 (1086, 217, 5),
 (1086, 218, 5), (1087, 105, 5), (1087, 198, 5), (1087, 199, 5), (1088, 195, 5), (1088, 220, 5),
 (1089, 104, 5), (1089, 200, 5), (1090, 103, 5), (1090, 388, 5), (1091, 198, 5), (1091, 370, 5),
 (1092, 365, 5), (1092, 375, 5), (1093, 334, 5), (1093, 354, 5), (1094, 360, 5), (1094, 361, 5),
 (1094, 363, 5),
 (1095, 28, 5), (1096, 44, 5), (1096, 53, 5), (1097, 218, 5), (1097, 219, 5), (1097, 362, 5), (1097, 364, 5),
 (1098, 217, 5), (1098, 362, 5), (1098, 351, 5), (1099, 353, 5), (1099, 361, 5), (1100, 380, 5),
 (1100, 381, 5), (1100, 385, 5), (1101, 199, 5), (1101, 200, 5), (1102, 359, 5), (1103, 387, 5),

 (1103, 388, 5), (1104, 399, 5), (1104, 402, 5), (1105, 71, 5), (1105, 72, 5), (1105, 127, 5), (1106, 60, 5),
 (1106, 62, 5), (1106, 71, 5), (1107, 59, 5), (1107, 63, 5), (1107, 64, 5), (1108, 108, 5), (1108, 110, 5),
 (1108, 111, 5), (1109, 107, 5), (1109, 108, 5), (1110, 68, 5), (1110, 70, 5), (1110, 81, 5),
 (1110, 119, 5), (1111, 80, 5), (1111, 98, 5), (1112, 65, 5), (1112, 110, 5), (1113, 105, 5), (1113, 106, 5),
 (1113, 107, 5), (1114, 34, 5), (1114, 85, 5), (1114, 86, 5), (1115, 87, 5), (1115, 95, 5), (1116, 86, 5),
 (1116, 90, 5), (1117, 33, 5), (1117, 90, 5), (1117, 92, 5), (1118, 89, 5), (1118, 91, 5), (1118, 92, 5),
 (1119, 91, 5), (1119, 93, 5), (1120, 151, 5), (1121, 402, 5), (1121, 402, 5), (1121, 414, 5), (1121, 415, 5),
 (1122, 319, 5), (1123, 385, 5), (1123, 386, 5), (1124, 32, 5), (1125, 403, 5), (1125, 404, 5), (1125, 413, 5),
 (1126, 96, 5), (1126, 97, 5), (1127, 419, 5), (1127, 423, 5), (1128, 400, 5), (1128, 417, 5),
 (1129, 417, 5), (1130, 417, 5), (1131, 416, 5), (1131, 418, 5), (1131, 419, 5), (1131, 419, 5), (1132, 406, 5),
 (1132, 408, 5), (1133, 411, 5), (1133, 422, 5), (1134, 16, 5), (1134, 24, 5), (1134, 44, 5), (1135, 56, 5),
 (1135, 76, 5), (1136, 54, 5), (1136, 75, 5), (1131, 419, 5), (1137, 52, 5), (1137, 53, 5),
 (1137, 74, 5), (1138, 47, 5), (1138, 48, 5), (1138, 52, 5), (1139, 51, 5), (1139, 52, 5), (1131, 71, 5),
 (1140, 51, 5), (1140, 52, 5), (1141, 48, 5), (1141, 50, 5), (1142, 38, 5), (1142, 24, 5), (1143, 38, 5),
 (1143, 39, 5), (1143, 40, 5), (1143, 42, 5), (1144, 34, 5), (1144, 67, 5), (1145, 74, 5), (1146, 45, 5),

(1146,44,5), (1147,53,5), (1147,54,5), (1148,148,5), (1148,246,5), (1149,46,5), (1149,53,5), (1150,148,5), (1150,247,5), (1150,247,5), (1151,24,5), (1151,25,5), (1152,43,5), (1152,44,5), (1153,12,5), (1153,14,5), (1154,44,5), (1155,55,5), (1155,56,5), (1155,57,5), (1156,18,5),

(1156,22,5), (1157,9,5), (1157,20,5), (1158,20,5), (1158,23,5), (1158,24,5), (1159,26,5), (1159,27,5), (1160,51,5), (1160,60,5), (1160,71,5), (1161,75,5), (1161,76,5), (1162,10,5), (1162,25,5), (1162,26,5), (1163,5,5), (1163,8,5), (1163,7,5), (1164,359,5),

(77, 1001, 5), (143, 1002, 5), (145, 1002, 5), (142, 1002, 5), (141, 1003, 5), (75, 1004, 5), (144, 1005, 5), (146, 1005, 5), (153, 1006, 5), (139, 1007, 5), (153, 1007, 5), (130, 1007, 5), (144, 1008, 5), (146, 1008, 5), (147, 1009, 5), (149, 1010, 5), (153, 1010, 5), (154, 1011, 5), (165, 1012, 5), (164, 1012, 5), (149, 1013, 5), (164, 1014, 5), (145, 1015, 5), (140, 1015, 5), (343, 1016, 5), (340, 1016, 5), (263, 1017, 5), (264, 1017, 5), (257, 1018, 5), (257, 1019, 5), (255, 1019, 5), (252, 1020, 5), (248, 1020, 5), (240, 1021, 5), (266, 1021, 5), (246, 1022, 5), (246, 1023, 5), (248, 1023, 5), (242, 1024, 5), (243, 1025, 5), (250, 1026, 5), (264, 1027, 5), (265, 1027, 5), (241, 1028, 5), (242, 1028, 5), (235, 1029, 5), (236, 1029, 5), (235, 1030, 5), (236, 1030, 5), (234, 1031, 5), (237, 1031, 5), (234, 1032, 5), (301, 1033, 5), (303, 1033, 5), (305, 1034, 5), (306, 1034, 5), (233, 1035, 5), (308, 1036, 5), (309, 1036, 5), (286, 1037, 5), (237, 1038, 5), (278, 1039, 5), (292, 1039, 5), (271, 1040, 5), (148, 1041, 5), (158, 1042, 5), (159, 1042, 5), (159, 1043, 5), (169, 1044, 5), (170, 1044, 5), (222, 1045, 5), (157, 1046, 5), (161, 1046, 5), (349, 1047, 5), (350, 1047, 5), (229, 1048, 5), (231, 1048, 5), (322, 1049, 5), (323, 1049, 5), (342, 1050, 5), (343, 1050, 5), (159, 1051, 5), (232, 1051, 5), (231, 1052, 5), (232, 1052, 5), (347, 1053, 5), (346, 1053, 5), (347, 1053, 5), (336, 1054, 5), (331, 1055, 5), (337, 1055, 5), (318, 1056, 5), (320, 1056, 5), (322, 1056, 5), (352, 1057, 5), (353, 1057, 5), (354, 1057, 5), (251, 1058, 5), (353, 1059, 5), (354, 1059, 5), (149, 1060, 5), (150, 1060, 5), (353, 1061, 5), (354, 1061, 5), (329, 1062, 5), (315, 1063, 5), (330, 1063, 5), (359, 1064, 5), (317, 1065, 5), (341, 1065, 5), (329, 1066, 5), (129, 1067, 5), (131, 1067, 5), (186, 1068, 5), (121, 1068, 5), (39, 1069, 5), (65, 1069, 5), (66, 1069, 5), (114, 1070, 5), (115, 1070, 5), (120, 1070, 5), (121, 1070, 5), (152, 1071, 5), (172, 1072, 5), (175, 1072, 5), (167, 1073, 5), (223, 1073, 5), (187, 1074, 5), (191, 1074, 5), (192, 1074, 5), (189, 1075, 5), (190, 1075, 5), (194, 1075, 5), (193, 1076, 5), (202, 1076, 5), (206, 1076, 5), (417, 1077, 5), (194, 1078, 5), (206, 1078, 5), (152, 1079, 5), (157, 1079, 5), (180, 1080, 5), (210, 1080, 5), (181, 1081, 5), (183, 1081, 5), (207, 1082, 5), (219, 1082, 5), (210, 1083, 5), (214, 1083, 5), (123, 1084, 5), (174, 1084, 5), (214, 1085, 5), (216, 1085, 5), (351, 1085, 5), (207, 1086, 5), (217, 1086, 5), (218, 1086, 5), (105, 1087, 5), (198, 1087, 5), (199, 1087, 5), (195, 1088, 5), (220, 1088, 5), (104, 1089, 5), (200, 1089, 5), (103, 1090, 5), (388, 1090, 5), (198, 1091, 5), (370, 1091, 5), (365, 1092, 5), (375, 1092, 5), (334, 1093, 5), (354, 1093, 5), (360, 1094, 5), (361, 1094, 5), (363, 1094, 5), (28, 1095, 5), (44, 1096, 5), (53, 1096, 5), (218, 1097, 5), (219, 1097, 5), (362, 1097, 5), (364, 1097, 5), (217, 1098, 5), (362, 1098, 5), (351, 1098, 5), (353, 1099, 5), (361, 1099, 5), (380, 1100, 5), (381, 1100, 5), (385, 1100, 5), (199, 1101, 5), (200, 1101, 5), (359, 1102, 5), (387, 1103, 5), (388, 1103, 5), (399, 1104, 5), (402, 1104, 5), (71, 1105, 5), (72, 1105, 5), (127, 1105, 5), (60, 1106, 5), (62, 1106, 5), (71, 1106, 5), (59, 1107, 5), (63, 1107, 5), (64, 1107, 5), (108, 1108, 5), (110, 1108, 5), (111, 1108, 5), (107, 1109, 5), (108, 1109, 5), (68, 1110, 5), (70, 1110, 5), (81, 1110, 5), (119, 1110, 5), (80, 1111, 5), (98,

```

1111, 5), (65, 1112, 5), (110, 1112, 5), (105, 1113, 5), (106, 1113, 5),
(107, 1113, 5), (34, 1114, 5), (85, 1114, 5), (86, 1114, 5), (87, 1115, 5),
(95, 1115, 5), (86, 1116, 5), (90, 1116, 5), (33, 1117, 5), (90, 1117, 5),
(92, 1117, 5), (89, 1118, 5), (91, 1118, 5), (92, 1118, 5), (91, 1119, 5),
(93, 1119, 5), (151, 1120, 5), (402, 1121, 5), (402, 1121, 5), (414, 1121,
5), (415, 1121, 5), (319, 1122, 5), (385, 1123, 5), (386, 1123, 5), (32,
1124, 5), (403, 1125, 5), (404, 1125, 5), (413, 1125, 5), (96, 1126, 5), (97,
1126, 5), (419, 1127, 5), (423, 1127, 5), (400, 1128, 5), (417, 1128, 5),
(417, 1129, 5), (417, 1130, 5), (416, 1131, 5), (418, 1131, 5), (419, 1131,
5), (419, 1131, 5), (406, 1132, 5), (408, 1132, 5), (411, 1133, 5), (422,
1133, 5), (16, 1134, 5), (24, 1134, 5), (44, 1134, 5), (56, 1135, 5), (76,
1135, 5), (54, 1136, 5), (75, 1136, 5), (419, 1131, 5), (52, 1137, 5), (53,
1137, 5), (74, 1137, 5), (47, 1138, 5), (48, 1138, 5), (52, 1138, 5), (51,
1139, 5), (52, 1139, 5), (71, 1131, 5), (51, 1140, 5), (52, 1140, 5), (48,
1141, 5), (50, 1141, 5), (38, 1142, 5), (24, 1142, 5), (38, 1143, 5), (39,
1143, 5), (40, 1143, 5), (42, 1143, 5), (34, 1144, 5), (67, 1144, 5), (74,
1145, 5), (45, 1146, 5), (44, 1146, 5), (53, 1147, 5), (54, 1147, 5), (148,
1148, 5), (246, 1148, 5), (46, 1149, 5), (53, 1149, 5), (148, 1150, 5), (247,
1150, 5), (247, 1150, 5), (24, 1151, 5), (25, 1151, 5), (43, 1152, 5), (44,
1152, 5), (12, 1153, 5), (14, 1153, 5), (44, 1154, 5), (55, 1155, 5), (56,
1155, 5), (57, 1155, 5), (18, 1156, 5), (22, 1156, 5), (9, 1157, 5), (20,
1157, 5), (20, 1158, 5), (23, 1158, 5), (24, 1158, 5), (26, 1159, 5), (27,
1159, 5), (51, 1160, 5), (60, 1160, 5), (71, 1160, 5), (75, 1161, 5), (76,
1161, 5), (10, 1162, 5), (25, 1162, 5), (26, 1162, 5), (5, 1163, 5),
(8,1163,5), (7, 1163, 5), (359, 1164, 5)]

```

```
Hoja2='Hoja3'
```

```
Hoja3='Hoja5'
```

```
Rangos2={
```

```
  'Centroides':['W2','Y166'],
```

```
  'Conectores':['R3','U339'],
```

```
  'Demanda':['A1','FI165']
```

```
}
```

```
Centroides=Read_Excel_to_NesteDic(Book,Hoja2, Rangos2['Centroides'][0],
```

```
Rangos2['Centroides'][1])
```

```
ListaCentroides=[i for i in Centroides]
```

```
Demanda=Read_Excel_to_NesteDic(Book,Hoja3, Rangos2['Demanda'][0],
```

```
Rangos2['Demanda'][1])
```

```
# Inicializa una variable para almacenar los datos convertidos
```

```
Demanda_datos = {}
```

```
# Recorre el diccionario original
```

```
for i, valores in Demanda.items():
```

```
    for j, valor in valores.items():
```

```
        lista = (i, j)
```

```
        Demanda_datos[lista] = valor
```

```
# Imprime los Demanda_datos
```

```
#for tupla, valor in Demanda_datos.items():
```

```
    # print(tupla, valor)
```

```
SalidasDeCentroide={}
```

```

LlegadasACentrodie={}

for i in ListaCentroides:
    SalidasDeCentroide[i]=[]
    LlegadasACentrodie[i]=[]
    for (j,k,l) in Listaconectores:
        if i==j:
            SalidasDeCentroide[i].append((j,k,l))
        if k==i:
            LlegadasACentrodie[i].append((j,k,l))

def devuelve_arcos(ListaNodos):
    ListaArcos=[]
    for o,i in enumerate(ListaNodos[0:-1]):
        ListaArcos.append((ListaNodos[o],ListaNodos[o+1]))
    return ListaArcos

PathNodos={}
PathArcos={}
NumPathPorArco={}
DemandaPorArco={}
PathsQuePasanPorArco={}

for (i,j,k) in ListaArcos:
    NumPathPorArco[(i,j)] = 0
    DemandaPorArco[(i,j)] = 0
    PathsQuePasanPorArco[(i,j)]=[]

G2=nx.DiGraph()
G2.add_nodes_from(ListaNodos)
G2.add_weighted_edges_from(ListaArcos)

for i in ListaCentroides:
    for j in ListaCentroides:
        if i!=j:
            GAmp1=G2.copy()
            GAmp1.add_nodes_from([i,j])
            GAmp1.add_weighted_edges_from(SalidasDeCentroide[i])
            GAmp1.add_weighted_edges_from(LlegadasACentrodie[j])
            PathNodos[(i,j)]=nx.shortest_path(GAmp1, source=i, target=j,
weight='weight')
            PathArcos[(i, j)] = devuelve_arcos(PathNodos[(i,j)])

for (i,j) in PathArcos:
    for (a,b) in PathArcos[(i,j)]:
        if a!=i and b!=j:

```

```

NumPathPorArco[(a, b)]+=1
#PathsQuePasanPorArco[(a,b)].append((i,j))
DemandaPorArco[(a,b)]+=Demanda_datos[(i,j)]

```

```

rankingfp = sorted(DemandaPorArco.items(), key=lambda x: x[1],
reverse=True)#ranking flujo de personas
# Imprimir el ranking
print("Ranking:")
for i, (arco, personas) in enumerate(rankingfp, 1):
    print(personas)

```

```

Tcaminos=sum(NumPathPorArco.values())#calculo el numero total de caminos
print(Tcaminos)
print(sum(Demanda_datos.values()))#calculo el numero total de personas en la
red

```

```

suma_arcos = {}
# Iterar sobre el diccionario y sumar los valores
for arco, valor in NumPathPorArco.items():
    # Ordenar el par
    arco_ordenado = tuple(sorted(arco))

    # Sumar el valor al diccionario
    suma_arcos[arco_ordenado] = suma_arcos.get(arco_ordenado, 0) + valor

# Lista de arcos con el numero de caminos
for arco, suma in suma_arcos.items():
    print(f"{arco}: {suma}")

```

```

print('a')
print('a')

```

```

rankingc = sorted(suma_arcos.items(), key=lambda x: x[1], reverse=True)
#Imprimir el ranking de numeros de caminos por arco
#for i, (arco, caminos) in enumerate(rankingc):
    #print(caminos)#suma de caminos (a,b)+(b,a)

```

```

#print(NumPathPorArco)
print('a')
#print(PathsQuePasanPorArco)
#print(DemandaPorArco)
print('a')

```

```

suma_arcos2 = {}
# Iterar sobre el diccionario y sumar los valores
for arco, valor in DemandaPorArco.items():
    # Ordenar el par

```

```

arco_ordenado = tuple(sorted(arco))

# Sumar el valor al diccionario
suma_arcos2[arco_ordenado] = suma_arcos2.get(arco_ordenado, 0) + valor

# Lista de arcos con el numero de caminos
for arco, suma in suma_arcos2.items():
    print(f"{arco}: {suma}")

rankinap = sorted(suma_arcos2.items(), key=lambda x: x[1], reverse=True)
#for i, (arco, personas) in enumerate(rankinap):
    #print(arco, personas) #suma de arcos (a,b)+(b,a)

# Inicializar el diccionario para almacenar el número de personas que pasan
por cada nodo
personas_por_nodo = {}

# Iterar sobre la lista de arcos y sumar las cantidades de personas a cada
nodo
for (origen, destino), personas in DemandaPorArco.items():
    if origen not in personas_por_nodo:
        personas_por_nodo[origen] = 0
    if destino not in personas_por_nodo:
        personas_por_nodo[destino] = 0

    personas_por_nodo[origen] += personas
    personas_por_nodo[destino] += personas

# Imprimir nodo con sus respectivos flujos
for nodo, total_personas in personas_por_nodo.items():
    print(f"Nodo {nodo}: {total_personas} personas")

rankinnp = sorted(personas_por_nodo.items(), key=lambda x: x[1],
reverse=True)
for i, (nodo, personas) in enumerate(rankinnp):
    print(personas)

```

Programa 2. Gráficos y Visualizaciones.

```
from IOfunctionsExcel5 import *
import math
import networkx as nx
from matplotlib import pyplot as plt

Book='Red sevilla ampliada.xlsx'
Hoja='Hoja1'

Rangos={
'Nodos':['A2','C436'],
'Arcos':['H2','0718'],
'EscalaN1':['W4'],
'EscalaN2':['W6'],
'Escala':['W8'],

}

Nodos=Read_Excel_to_NesteDic(Book,Hoja, Rangos['Nodos'][0],
Rangos['Nodos'][1])
Arcos=Read_Excel_to_NesteDic_TupleInt(Book,Hoja, Rangos['Arcos'][0],
Rangos['Arcos'][1])
Nodo1=Read_Excel_Cell(Book,Hoja, Rangos['EscalaN1'][0])
Nodo2=Read_Excel_Cell(Book,Hoja, Rangos['EscalaN2'][0])
DistReal_N1N2=Read_Excel_Cell(Book,Hoja, Rangos['Escala'][0])
Distancia_N1_N2=math.sqrt((Nodos[Nodo1]['X']-
Nodos[Nodo2]['X'])**2+(Nodos[Nodo1]['Y']-Nodos[Nodo2]['Y'])**2 )

Factor_Escala= DistReal_N1N2/Distancia_N1_N2

#Para trabajar con arcos no duplicados
Arcos2=Read_Excel_to_NesteDic_TupleInt(Book,Hoja, Rangos['Arcos'][0],
Rangos['Arcos'][1])

#Si quisieramos duplicar arcos con doble sentido

Duplicados={}
for (i,j) in Arcos:
    if Arcos[(i,j)]['Sentidos']==2:
        Duplicados.update({(j,i):{'Origen':j, 'Destino':i,
'Factor':Arcos[(i,j)]['Factor'],
'Tipo':Arcos[(i,j)]['Tipo'],'Velocidad':Arcos[(i,j)]['Velocidad'],
'Sentidos':Arcos[(i,j)]['Sentidos'],'carriles':Arcos[(i,j)]['carriles'] }})

Arcos.update(Duplicados)

# Crear longitudes calculadas de todos Los arcos

Longitudes={}
tiempo={}#para La segunda parte
```

```

for (i,j) in Arcos:
    if Arcos[(i,j)]['Factor']==1:
        Longitudes[(i,j)]=Factor_Escala* math.sqrt((Nodos[i]['X']-
Nodos[j]['X'])**2+(Nodos[i]['Y']-Nodos[j]['Y'])**2 )
        tiempo[(i, j)] = 60 * (((Longitudes[(i, j)])) / 1000) / Arcos[(i,
j)]['Velocidad'] # para la segunda parte
    else:
        Longitudes[(i,j)]=Arcos[(i,j)]['Factor'];
        tiempo[(i, j)] = 60 * (((Longitudes[(i, j)])) / 1000) / Arcos[(i,
j)]['Velocidad'] # para la segunda parte

Tipo_velocidad={}

for (i,j) in Arcos:
    Tipo_velocidad[(i,j)]= Arcos[(i,j)]['Velocidad'] #en verde hayque ponerel
nombre que venga en la columna del excel

G=nx.DiGraph()
ListaNodos=[i for i in Nodos]
ListaArcos=[(i,j,tiempo[(i,j)]) for (i,j) in Arcos]
G.add_nodes_from(ListaNodos)
G.add_weighted_edges_from(ListaArcos)
NodosPosiciones={i:[Nodos[i]['X'],Nodos[i]['Y']] for i in Nodos}

nx.set_edge_attributes(G, Longitudes, name='Long')

#Eliminar los nodos o arcos
#eliminar_arcos(grafo_copia, arcos_a_eliminar )
#funcion elimina arcos
def eliminar_arcos(grafo, arc_list):
    for arco in arc_list:
        if grafo.has_edge(*arco): #Arcos es una tupla
            grafo.remove_edge(*arco)

#funcion elimina nodos
def eliminar_nodos(grafo, nodo_list):
    grafo.remove_nodes_from(nodo_list)

#nodos_a_eliminar = [10, 300]
arcos_a_eliminar = [(27, 28), (28, 29), (28,27), (29,28)]
grafo_copia = G.copy()

NetProperties = {'NodeSize': 10, 'NodeColor': 'blue', 'Transparency': None,
                'NodeBorderColor': 'black', 'NodeBorderWidth': 0.2,
                'NodeShape': 'o',
                'LinkWidth':0.5, 'LinksColor': 'grey',
                'Labels': False, 'FontColor': 'black', 'FontSize': 10,

```

```

        'Arrows': False, 'ArrowSize': 0.0,
        'BackgroundLinkColor': 'grey', 'BackgroundLinkWidth': 0.3,
'BackgroundNodeBorderColor': 'grey', 'BackgroundNodeBorderWidth': 0.1,
        'BackgroundNodeColor': 'grey', 'BackgroundNodeSize': 2,
'BackgroundNodeShape': 'o',
        'NoAxes': True, 'TightLayout': True,
    }

```

Closeness Centrality

```

Cl_centrality=nx.closeness_centrality(G, distance='long')
MaxCl=max([Cl_centrality[i] for i in Cl_centrality])
Scale_centrality=50
Sizes=[Scale_centrality*(Cl_centrality[i]/MaxCl) for i in Cl_centrality]
Transparency=[(Cl_centrality[i]/MaxCl) for i in Cl_centrality]
print('Centralities', Cl_centrality)#medidas topológicas

```

```

fig, ax = plt.subplots()
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(grafo_copia, pos=NodosPosiciones, node_size=Sizes,
with_labels=NetProperties['Labels'],
                font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
                node_shape=NetProperties['NodeShape'],
                width=NetProperties['LinkWidth'],
arrows=NetProperties['Arrows'],
                edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
                linewidths=NetProperties['NodeBorderWidth'],
alpha=Transparency)
#nx.draw_networkx(grafo_copia, NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Closeness Centrality', fontsize= 10)
plt.show()

```

```

CarrilesEntrantes={}
CarrilesSalientes={}
for node in G.nodes():
    CarrilesEntrantes[node] = 0
    CarrilesSalientes[node] = 0
    outedges = list(G.out_edges(node)) # proporciona el número de arcos que
salen de un nodo
    for (i,j) in outedges:
        CarrilesSalientes[node]+=Arcos[(i,j)][ 'carriles' ]
    inedges = list(G.in_edges(node)) #proporciona el número de arcos que
entran en un nodo
    for (i,j) in inedges:
        CarrilesEntrantes[node]+=Arcos[(i,j)][ 'carriles' ]

```



```

# INDEGREE CARRILES
Scale_indegree=10
MaxIn=max([CarrilesEntrantes[i] for i in Cl_centrality])
Transparency=[CarrilesEntrantes[i]/MaxIn for i in CarrilesEntrantes]
Sizes=[Scale_indegree*CarrilesEntrantes[i] for i in CarrilesEntrantes]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(grafo_copia, pos=NodosPosiciones, node_size=Sizes,
with_labels=NetProperties['Labels'],
font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
node_shape=NetProperties['NodeShape'],
width=NetProperties['LinkWidth'],
arrows=NetProperties['Arrows'],
edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
linewidths=NetProperties['NodeBorderWidth'],
alpha=Transparency)
#nx.draw_networkx(grafo_copia,NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Indegree with lanes', fontsize=10)
plt.show()

# OUTDEGREE CARRILES
Scale_outdegree=Scale_indegree
MaxIn=max([CarrilesSalientes[i] for i in CarrilesSalientes])
Transparency=[CarrilesSalientes[i]/MaxIn for i in CarrilesSalientes]
Sizes=[Scale_outdegree*CarrilesSalientes[i] for i in CarrilesSalientes]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(grafo_copia, pos=NodosPosiciones, node_size=Sizes,
with_labels=NetProperties['Labels'],
font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
node_shape=NetProperties['NodeShape'],
width=NetProperties['LinkWidth'],
arrows=NetProperties['Arrows'],
edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
linewidths=NetProperties['NodeBorderWidth'],
alpha=Transparency)
#nx.draw_networkx(grafo_copia,NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Outdegree with lanes', fontsize=10)
plt.show()

```

```

for node in G.nodes():
    peso1=0
    outedges = list(G.out_edges(node)) #proporciona el numero de arcos que salen de un nodo
    inedges = list(G.in_edges(node)) #nos dice el numero de arcos que entran en un nodo
    print(f"Node {node}:")
    print(f"\tCarriles que llegan: {inedges}")
    print(f"\tCarriles que salen: {outedges}")
    print()
    for (i,j) in Arcos:
        if node == Arcos[(i, j)]['Destino'] and Arcos[(i,j)]['Sentidos']==2:
            peso1 =peso1+ Arcos[(i, j)]['carriles']*2
        else:
            if (node == Arcos[(i, j)]['Destino'] or node == Arcos[(i, j)]['Origen']) and Arcos[(i,j)]['Sentidos']==1:
                peso1 = peso1 + Arcos[(i, j)]['carriles']
    print('peso',peso1)

#para ver a que nodos no les llega ningun arco, solo salen de él
def nodo_sin_fuente(grafo, nodo):
    for n in G.nodes():
        if n != nodo and grafo.has_edge(n, nodo):
            return False
    return True

for nodo in G.nodes():
    if nodo_sin_fuente(G, nodo):
        print(nodo)

#ver la centralidad de un nodo
# Calcular la centralidad de cercanía para todos los nodos
#for node in G.nodes():
    #closeness centrality = nx.closeness centrality(G, distance='Long')
#distance= Longitudes[(i,j)], sale igual dandole valor 1 a cada distancia

# Los resultados de la centralidad de cercanía
#for nodo, valor in closeness centrality.items():
    #print(f"Centralidad de cercanía del nodo {nodo}: {valor}")

Between_Centrality=nx.betweenness centrality(G, weight='Long')
print('distance',Between_Centrality)#medida con distancias
MaxBetwCent=max([Between_Centrality[i] for i in Cl_centrality])
Scale_centrality=100
Sizes=[Scale_centrality*(Between_Centrality[i]/MaxBetwCent) for i in Between_Centrality]
Transparency=[(Between_Centrality[i]/MaxBetwCent) for i in Between_Centrality]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0, wspace=0)
ax.axis('equal')

```

```

ax.axis("off")
nx.draw_networkx(grafo_copia, pos=NodosPosiciones, node_size=Sizes,
with_labels=NetProperties['Labels'],
                font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
                node_shape=NetProperties['NodeShape'],
                width=NetProperties['LinkWidth'],
arrows=NetProperties['Arrows'],
                edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
                linewidths=NetProperties['NodeBorderWidth'],
alpha=Transparency)
#nx.draw_networkx(grafo_copia,NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Beteewnness Centrality', fontsize=10)
plt.show()

#importancia del viario

#Arcos[(i,j)]['Sentidos']==2:

CarrilesTotalesArcos={}
for (i,j) in Arcos2:
    CarrilesTotalesArcos[(i,j)]=Arcos2[(i,j)]['carriles']
    if Arcos2[(i,j)]['Sentidos']==2:
        CarrilesTotalesArcos[(i, j)]+=Arcos2[(i,j)]['carriles']
MaxCarriles=max([CarrilesTotalesArcos[(i, j)] for (i,j) in
CarrilesTotalesArcos])
Link_scale=10
Sizes=[Link_scale*(CarrilesTotalesArcos[(i, j)]/MaxCarriles) for (i,j) in
CarrilesTotalesArcos]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(grafo_copia, edgelist=Arcos2, pos=NodosPosiciones,
node_size=8, with_labels=NetProperties['Labels'],
                font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
                node_shape=NetProperties['NodeShape'],
                width=Sizes, arrows=NetProperties['Arrows'],
                edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
                linewidths=NetProperties['NodeBorderWidth'])
#nx.draw_networkx(grafo_copia,NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Total number of lanes', fontsize=10)
plt.show()

#Lo añado a lo del profesor
Tipo_velocidad={}

for (i,j) in Arcos:

```

```

    Tipo_velocidad[(i,j)]= Arcos[(i,j)]['Velocidad']
Tipo_velocidad={}
for (i,j) in Arcos:
    Tipo_velocidad[(i,j)]= Arcos[(i,j)]['Velocidad']
Link_scale=5
Sizes=[Link_scale*(Tipo_velocidad[(i,j)]/80) for (i,j) in Tipo_velocidad]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(grafo_copia, edgelist=Arcos2, pos=NodosPosiciones,
node_size=8, with_labels=NetProperties['Labels'],
                    font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
                    node_shape=NetProperties['NodeShape'],
                    width=Sizes, arrows=NetProperties['Arrows'],
                    edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
                    linewidths=NetProperties['NodeBorderWidth'])
#nx.draw_networkx(grafo_copia,NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Velocity', fontsize=10)
plt.show()#hasta aqui lo que yo he añadido

```

```

G2=nx.DiGraph()
ListaNodos=[i for i in Nodos]
ListaArcos2=[(i,j,Longitudes[(i,j)]) for (i,j) in Arcos2]
G2.add_nodes_from(ListaNodos)
G2.add_weighted_edges_from(ListaArcos2)
Sizes=[CarrilesTotalesArcos[(i, j)] for (i,j) in CarrilesTotalesArcos]
nx.set_edge_attributes(G2, Sizes, name='NumCarriles')

```

```

# Calcular el PageRank
pagerank = nx.pagerank(G, alpha=0.85, max_iter=100, tol=1e-06) #,
weight='NumCarriles'
MaxPageRank=max([pagerank[i] for i in pagerank])
Scale_centrality=100
Sizes=[Scale_centrality*(pagerank[i]/MaxPageRank) for i in pagerank]
Transparency=[(pagerank[i]/MaxPageRank) for i in pagerank]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(G, pos=NodosPosiciones, node_size=Sizes,
with_labels=NetProperties['Labels'],
                    font_size=NetProperties['FontSize'],

```

```

node_color=NetProperties['NodeColor'],
        node_shape=NetProperties['NodeShape'],
        width=NetProperties['LinkWidth'],
arrows=NetProperties['Arrows'],
        edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
        linewidths=NetProperties['NodeBorderWidth'],
alpha=Transparency)
#nx.draw_networkx(grafo_copia,NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('PageRank', fontsize=10)
plt.show()

# Imprimo Los valores de PageRank para cada nodo
#for node, score in pagerank.items():
    #print(f"Node {node}: PageRank = {score}")

# MEDIDAS CON DISTANCIAS

print(nx.betweenness_centrality(G))#medidas topológicas

print('distance',nx.closeness_centrality(G, distance='Long'))#medida con
distancias
print(nx.closeness_centrality(G)) #medida topologica

fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(G,NodosPosiciones, with_labels=False, node_size=5,
font_size=6, arrowstyle='-') #with_labels=True,
plt.title('Base network', fontsize=10)
plt.show()
#print(a)

#segundo parte, estudio con flujo de personas y tiempo entre arcos
Listaconectores=[(1001,77,5),(1002,143,5),(1002,145,5),(1002,142,5),(1003,141
,5),(1004,75,5),(1005,144,5),(1005,146,5),(1006,153,5),(1007,139,5),(1007,153
,5),(1007,130,5),(1008,144,5),(1008,146,5),(1009,147,5),
(1010,149,5),(1010,153,5),(1011,154,5),(1012,165,5),(1012,164,5),(1013,149,5)
,(1014,164,5),(1015,145,5),(1015,140,5),(1016,343,5),(1016,340,5),(1017,263,5
),(1017,264,5),(1018,257,5),(1019,257,5),(1019,255,5),(1020,252,5),(1020,248,
5),
(1021,240,5),(1021,266,5),(1022,246,5),(1023,246,5),(1023,248,5),(1024,242,5)
,(1025,243,5),(1026,250,5),(1027,264,5),(1027,265,5),(1028,241,5),(1028,242,5
),(1029,235,5),(1029,236,5),(1030,235,5),(1030,236,5),
(1031,234,5),(1031,237,5),(1032,234,5),(1033,301,5),(1033,303,5),(1034,305,5)

```

, (1034, 306, 5), (1035, 233, 5), (1036, 308, 5), (1036, 309, 5), (1037, 286, 5), (1038, 237, 5), (1039, 278, 5), (1039, 292, 5), (1040, 271, 5), (1041, 148, 5), (1042, 158, 5), (1042, 159, 5), (1043, 159, 5), (1044, 169, 5), (1044, 170, 5), (1045, 222, 5), (1046, 157, 5), (1046, 161, 5), (1047, 349, 5), (1047, 350, 5), (1048, 229, 5), (1048, 231, 5), (1049, 322, 5), (1049, 323, 5), (1050, 342, 5), (1050, 343, 5), (1051, 159, 5), (1051, 232, 5), (1052, 231, 5), (1052, 232, 5), (1053, 347, 5), (1053, 346, 5), (1053, 347, 5), (1054, 336, 5), (1055, 331, 5), (1055, 337, 5), (1056, 318, 5), (1056, 320, 5), (1056, 322, 5), (1057, 352, 5), (1057, 353, 5), (1057, 354, 5), (1058, 251, 5), (1059, 353, 5), (1059, 354, 5), (1060, 149, 5), (1060, 150, 5), (1061, 353, 5), (1061, 354, 5), (1062, 329, 5), (1063, 315, 5), (1063, 330, 5), (1064, 359, 5), (1065, 317, 5), (1065, 341, 5), (1066, 329, 5), (1067, 129, 5), (1067, 131, 5), (1068, 186, 5), (1068, 121, 5), (1069, 39, 5), (1069, 65, 5), (1069, 66, 5), (1070, 114, 5), (1070, 115, 5), (1070, 120, 5), (1070, 121, 5), (1071, 152, 5), (1072, 172, 5), (1072, 175, 5), (1073, 167, 5), (1073, 223, 5), (1074, 187, 5), (1074, 191, 5), (1074, 192, 5), (1075, 189, 5), (1075, 190, 5), (1075, 194, 5), (1076, 193, 5), (1076, 202, 5), (1076, 206, 5), (1077, 417, 5), (1078, 194, 5), (1078, 206, 5), (1079, 152, 5), (1079, 157, 5), (1080, 180, 5), (1080, 210, 5), (1081, 181, 5), (1081, 183, 5), (1082, 207, 5), (1082, 219, 5), (1083, 210, 5), (1083, 214, 5), (1084, 123, 5), (1084, 174, 5), (1085, 214, 5), (1085, 216, 5), (1085, 351, 5), (1086, 207, 5), (1086, 217, 5), (1086, 218, 5), (1087, 105, 5), (1087, 198, 5), (1087, 199, 5), (1088, 195, 5), (1088, 220, 5), (1089, 104, 5), (1089, 200, 5), (1090, 103, 5), (1090, 388, 5), (1091, 198, 5), (1091, 370, 5), (1092, 365, 5), (1092, 375, 5), (1093, 334, 5), (1093, 354, 5), (1094, 360, 5), (1094, 361, 5), (1094, 363, 5), (1095, 28, 5), (1096, 44, 5), (1096, 53, 5), (1097, 218, 5), (1097, 219, 5), (1097, 362, 5), (1097, 364, 5), (1098, 217, 5), (1098, 362, 5), (1098, 351, 5), (1099, 353, 5), (1099, 361, 5), (1100, 380, 5), (1100, 381, 5), (1100, 385, 5), (1101, 199, 5), (1101, 200, 5), (1102, 359, 5), (1103, 387, 5), (1103, 388, 5), (1104, 399, 5), (1104, 402, 5), (1105, 71, 5), (1105, 72, 5), (1105, 127, 5), (1106, 60, 5), (1106, 62, 5), (1106, 71, 5), (1107, 59, 5), (1107, 63, 5), (1107, 64, 5), (1108, 108, 5), (1108, 110, 5), (1108, 111, 5), (1109, 107, 5), (1109, 108, 5), (1110, 68, 5), (1110, 70, 5), (1110, 81, 5), (1110, 119, 5), (1111, 80, 5), (1111, 98, 5), (1112, 65, 5), (1112, 110, 5), (1113, 105, 5), (1113, 106, 5), (1113, 107, 5), (1114, 34, 5), (1114, 85, 5), (1114, 86, 5), (1115, 87, 5), (1115, 95, 5), (1116, 86, 5), (1116, 90, 5), (1117, 33, 5), (1117, 90, 5), (1117, 92, 5), (1118, 89, 5), (1118, 91, 5), (1118, 92, 5), (1119, 91, 5), (1119, 93, 5), (1120, 151, 5), (1121, 402, 5), (1121, 402, 5), (1121, 414, 5), (1121, 415, 5), (1122, 319, 5), (1123, 385, 5), (1123, 386, 5), (1124, 32, 5), (1125, 403, 5), (1125, 404, 5), (1125, 413, 5), (1126, 96, 5), (1126, 97, 5), (1127, 419, 5), (1127, 423, 5), (1128, 400, 5), (1128, 417, 5), (1129, 417, 5), (1130, 417, 5), (1131, 416, 5), (1131, 418, 5), (1131, 419, 5), (1131, 419, 5), (1132, 406, 5), (1132, 408, 5), (1133, 411, 5), (1133, 422, 5), (1134, 16, 5), (1134, 24, 5), (1134, 44, 5), (1135, 56, 5), (1135, 76, 5), (1136, 54, 5), (1136, 75, 5), (1131, 419, 5), (1137, 52, 5), (1137, 53, 5), (1137, 74, 5), (1138, 47, 5), (1138, 48, 5), (1138, 52, 5), (1139, 51, 5), (1139, 52, 5), (1131, 71, 5), (1140, 51, 5), (1140, 52, 5), (1141, 48, 5), (1141, 50, 5), (1142, 38, 5), (1142, 24, 5), (1143, 38, 5), (1143, 39, 5), (1143, 40, 5), (1143, 42, 5), (1144, 34, 5), (1144, 67, 5), (1145, 74, 5), (1146, 45, 5), (1146, 44, 5), (1147, 53, 5), (1147, 54, 5), (1148, 148, 5), (1148, 246, 5), (1149, 46, 5), (1149, 53, 5), (1150, 148, 5), (1150, 247, 5), (1150, 247, 5), (1151, 24, 5), (1151, 25, 5), (1152, 43, 5), (1152, 44, 5), (1153, 12, 5), (1153, 14, 5), (1154, 44, 5), (1155, 55, 5), (1155, 56, 5), (1155, 57, 5), (1156, 18, 5), (1156, 22, 5), (1157, 9, 5), (1157, 20, 5), (1158, 20, 5), (1158, 23, 5), (1158, 24, 5), (1159,

26,5), (1159,27,5), (1160,51,5), (1160,60,5), (1160,71,5), (1161,75,5), (1161,76,5),
 , (1162,10,5), (1162,25,5), (1162,26,5), (1163,5,5), (1163,8,5), (1163,7,5), (1164,3
 59,5),
 (77, 1001, 5), (143, 1002, 5), (145, 1002, 5), (142, 1002, 5), (141, 1003,
 5), (75, 1004, 5), (144, 1005, 5), (146, 1005, 5), (153, 1006, 5), (139,
 1007, 5), (153, 1007, 5), (130, 1007, 5), (144, 1008, 5), (146, 1008, 5),
 (147, 1009, 5), (149, 1010, 5), (153, 1010, 5), (154, 1011, 5), (165, 1012,
 5), (164, 1012, 5), (149, 1013, 5), (164, 1014, 5), (145, 1015, 5), (140,
 1015, 5), (343, 1016, 5), (340, 1016, 5), (263, 1017, 5), (264, 1017, 5),
 (257, 1018, 5), (257, 1019, 5), (255, 1019, 5), (252, 1020, 5), (248, 1020,
 5), (240, 1021, 5), (266, 1021, 5), (246, 1022, 5), (246, 1023, 5), (248,
 1023, 5), (242, 1024, 5), (243, 1025, 5), (250, 1026, 5), (264, 1027, 5),
 (265, 1027, 5), (241, 1028, 5), (242, 1028, 5), (235, 1029, 5), (236, 1029,
 5), (235, 1030, 5), (236, 1030, 5), (234, 1031, 5), (237, 1031, 5), (234,
 1032, 5), (301, 1033, 5), (303, 1033, 5), (305, 1034, 5), (306, 1034, 5),
 (233, 1035, 5), (308, 1036, 5), (309, 1036, 5), (286, 1037, 5), (237, 1038,
 5), (278, 1039, 5), (292, 1039, 5), (271, 1040, 5), (148, 1041, 5), (158,
 1042, 5), (159, 1042, 5), (159, 1043, 5), (169, 1044, 5), (170, 1044, 5),
 (222, 1045, 5), (157, 1046, 5), (161, 1046, 5), (349, 1047, 5), (350, 1047,
 5), (229, 1048, 5), (231, 1048, 5), (322, 1049, 5), (323, 1049, 5), (342,
 1050, 5), (343, 1050, 5), (159, 1051, 5), (232, 1051, 5), (231, 1052, 5),
 (232, 1052, 5), (347, 1053, 5), (346, 1053, 5), (347, 1053, 5), (336, 1054,
 5), (331, 1055, 5), (337, 1055, 5), (318, 1056, 5), (320, 1056, 5), (322,
 1056, 5), (352, 1057, 5), (353, 1057, 5), (354, 1057, 5), (251, 1058, 5),
 (353, 1059, 5), (354, 1059, 5), (149, 1060, 5), (150, 1060, 5), (353, 1061,
 5), (354, 1061, 5), (329, 1062, 5), (315, 1063, 5), (330, 1063, 5), (359,
 1064, 5), (317, 1065, 5), (341, 1065, 5), (329, 1066, 5), (129, 1067, 5),
 (131, 1067, 5), (186, 1068, 5), (121, 1068, 5), (39, 1069, 5), (65, 1069, 5),
 (66, 1069, 5), (114, 1070, 5), (115, 1070, 5), (120, 1070, 5), (121, 1070,
 5), (152, 1071, 5), (172, 1072, 5), (175, 1072, 5), (167, 1073, 5), (223,
 1073, 5), (187, 1074, 5), (191, 1074, 5), (192, 1074, 5), (189, 1075, 5),
 (190, 1075, 5), (194, 1075, 5), (193, 1076, 5), (202, 1076, 5), (206, 1076,
 5), (417, 1077, 5), (194, 1078, 5), (206, 1078, 5), (152, 1079, 5), (157,
 1079, 5), (180, 1080, 5), (210, 1080, 5), (181, 1081, 5), (183, 1081, 5),
 (207, 1082, 5), (219, 1082, 5), (210, 1083, 5), (214, 1083, 5), (123, 1084,
 5), (174, 1084, 5), (214, 1085, 5), (216, 1085, 5), (351, 1085, 5), (207,
 1086, 5), (217, 1086, 5), (218, 1086, 5), (105, 1087, 5), (198, 1087, 5),
 (199, 1087, 5), (195, 1088, 5), (220, 1088, 5), (104, 1089, 5), (200, 1089,
 5), (103, 1090, 5), (388, 1090, 5), (198, 1091, 5), (370, 1091, 5), (365,
 1092, 5), (375, 1092, 5), (334, 1093, 5), (354, 1093, 5), (360, 1094, 5),
 (361, 1094, 5), (363, 1094, 5), (28, 1095, 5), (44, 1096, 5), (53, 1096, 5),
 (218, 1097, 5), (219, 1097, 5), (362, 1097, 5), (364, 1097, 5), (217, 1098,
 5), (362, 1098, 5), (351, 1098, 5), (353, 1099, 5), (361, 1099, 5), (380,
 1100, 5), (381, 1100, 5), (385, 1100, 5), (199, 1101, 5), (200, 1101, 5),
 (359, 1102, 5), (387, 1103, 5), (388, 1103, 5), (399, 1104, 5), (402, 1104,
 5), (71, 1105, 5), (72, 1105, 5), (127, 1105, 5), (60, 1106, 5), (62, 1106,
 5), (71, 1106, 5), (59, 1107, 5), (63, 1107, 5), (64, 1107, 5), (108, 1108,
 5), (110, 1108, 5), (111, 1108, 5), (107, 1109, 5), (108, 1109, 5), (68,
 1110, 5), (70, 1110, 5), (81, 1110, 5), (119, 1110, 5), (80, 1111, 5), (98,
 1111, 5), (65, 1112, 5), (110, 1112, 5), (105, 1113, 5), (106, 1113, 5),
 (107, 1113, 5), (34, 1114, 5), (85, 1114, 5), (86, 1114, 5), (87, 1115, 5),
 (95, 1115, 5), (86, 1116, 5), (90, 1116, 5), (33, 1117, 5), (90, 1117, 5),
 (92, 1117, 5), (89, 1118, 5), (91, 1118, 5), (92, 1118, 5), (91, 1119, 5),
 (93, 1119, 5), (151, 1120, 5), (402, 1121, 5), (402, 1121, 5), (414, 1121,
 5), (415, 1121, 5), (319, 1122, 5), (385, 1123, 5), (386, 1123, 5), (32,

```

1124, 5), (403, 1125, 5), (404, 1125, 5), (413, 1125, 5), (96, 1126, 5), (97,
1126, 5), (419, 1127, 5), (423, 1127, 5), (400, 1128, 5), (417, 1128, 5),
(417, 1129, 5), (417, 1130, 5), (416, 1131, 5), (418, 1131, 5), (419, 1131,
5), (419, 1131, 5), (406, 1132, 5), (408, 1132, 5), (411, 1133, 5), (422,
1133, 5), (16, 1134, 5), (24, 1134, 5), (44, 1134, 5), (56, 1135, 5), (76,
1135, 5), (54, 1136, 5), (75, 1136, 5), (419, 1131, 5), (52, 1137, 5), (53,
1137, 5), (74, 1137, 5), (47, 1138, 5), (48, 1138, 5), (52, 1138, 5), (51,
1139, 5), (52, 1139, 5), (71, 1131, 5), (51, 1140, 5), (52, 1140, 5), (48,
1141, 5), (50, 1141, 5), (38, 1142, 5), (24, 1142, 5), (38, 1143, 5), (39,
1143, 5), (40, 1143, 5), (42, 1143, 5), (34, 1144, 5), (67, 1144, 5), (74,
1145, 5), (45, 1146, 5), (44, 1146, 5), (53, 1147, 5), (54, 1147, 5), (148,
1148, 5), (246, 1148, 5), (46, 1149, 5), (53, 1149, 5), (148, 1150, 5), (247,
1150, 5), (247, 1150, 5), (24, 1151, 5), (25, 1151, 5), (43, 1152, 5), (44,
1152, 5), (12, 1153, 5), (14, 1153, 5), (44, 1154, 5), (55, 1155, 5), (56,
1155, 5), (57, 1155, 5), (18, 1156, 5), (22, 1156, 5), (9, 1157, 5), (20,
1157, 5), (20, 1158, 5), (23, 1158, 5), (24, 1158, 5), (26, 1159, 5), (27,
1159, 5), (51, 1160, 5), (60, 1160, 5), (71, 1160, 5), (75, 1161, 5), (76,
1161, 5), (10, 1162, 5), (25, 1162, 5), (26, 1162, 5), (5, 1163, 5),
(8,1163,5), (7, 1163, 5), (359, 1164, 5)]

```

```
Hoja2='Hoja3'
```

```
Hoja3='Hoja5'
```

```
Rangos2={
```

```
'Centroides':['W2','Y166'],
```

```
'Conectores':['R3','U339'],
```

```
'Demanda':['A1','FI165']
```

```
}
```

```
Centroides=Read_Excel_to_NesteDic(Book,Hoja2, Rangos2['Centroides'][0],
```

```
Rangos2['Centroides'][1])
```

```
ListaCentroides=[i for i in Centroides]
```

```
Demanda=Read_Excel_to_NesteDic(Book,Hoja3, Rangos2['Demanda'][0],
```

```
Rangos2['Demanda'][1])
```

```
# Inicializa una variable para almacenar los datos convertidos
```

```
Demanda_datos = {}
```

```
# Recorre el diccionario original
```

```
for i, valores in Demanda.items():
```

```
    for j, valor in valores.items():
```

```
        lista = (i, j)
```

```
        Demanda_datos[lista] = valor
```

```
# Imprime Los Demanda_datos
```

```
#for tupla, valor in Demanda_datos.items():
```

```
    # print(tupla, valor)
```

```
SalidasDeCentroide={}
```

```
LlegadasACentrodie={}
```

```
for i in ListaCentroides:
```

```
    SalidasDeCentroide[i]=[]
```

```
    LlegadasACentrodie[i]=[]
```

```
    for (j,k,l) in Listaconectores:
```



```

        if i==j:
            SalidasDeCentroide[i].append((j,k,l))
        if k==i:
            LlegadasACentrodie[i].append((j,k,l))

def devuelve_arcos(ListaNodos):
    ListaArcos=[]
    for o,i in enumerate(ListaNodos[0:-1]):
        ListaArcos.append((ListaNodos[o],ListaNodos[o+1]))
    return ListaArcos

PathNodos={}
PathArcos={}
NumPathPorArco={}
DemandaPorArco={}
PathsQuePasanPorArco={}

for (i,j,k) in ListaArcos:
    NumPathPorArco[(i,j)] = 0
    DemandaPorArco[(i,j)] = 0
    PathsQuePasanPorArco[(i,j)]=[]

G2=nx.DiGraph()
G2.add_nodes_from(ListaNodos)
G2.add_weighted_edges_from(ListaArcos)

for i in ListaCentroides:
    for j in ListaCentroides:
        if i!=j:
            GAmpl=G2.copy()
            GAmpl.add_nodes_from([i,j])
            GAmpl.add_weighted_edges_from(SalidasDeCentroide[i])
            GAmpl.add_weighted_edges_from(LlegadasACentrodie[j])
            PathNodos[(i,j)]=nx.shortest_path(GAmpl, source=i, target=j,
weight='weight')
            PathArcos[(i, j)] = devuelve_arcos(PathNodos[(i,j)])

for (i,j) in PathArcos:
    for (a,b) in PathArcos[(i,j)]:
        if a!=i and b!=j:
            NumPathPorArco[(a, b)]+=1
            #PathsQuePasanPorArco[(a,b)].append((i,j))
            DemandaPorArco[(a,b)]+=Demanda_datos[(i,j)]

```

```
Tcaminos=sum(NumPathPorArco.values())#calculo el numero total de caminos
#print(Tcaminos)
#print(sum(Demanda_datos.values()))#calculo el numero total de personas en la
red
```

```
suma_arcos = {}
# Iterar sobre el diccionario y sumar los valores
for arco, valor in NumPathPorArco.items():
    # Ordenar el par
    arco_ordenado = tuple(sorted(arco))

    # Sumar el valor al diccionario
    suma_arcos[arco_ordenado] = suma_arcos.get(arco_ordenado, 0) + valor

# Lista de arcos con el numero de caminos
#for arco, suma in suma_arcos.items():
    # print(f"{arco}: {suma}")
```

```
print('a')
print('a')
# Nodos que deseas eliminar
nodos_a_eliminar = [1164,1163]

# Eliminar los nodos
GAmp1.remove_nodos_from(nodos_a_eliminar)
MaxCaminos=max([NumPathPorArco[(i, j)] for (i,j) in NumPathPorArco])
Link_scale=10#probar
Sizes=[Link_scale*(NumPathPorArco[(i, j)]/MaxCaminos) for (i,j) in
NumPathPorArco]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(GAmp1, edgelist=Arcos2, pos=NodosPosiciones, node_size=8,
with_labels=NetProperties['Labels'],
                font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
                node_shape=NetProperties['NodeShape'],
                width=Sizes, arrows=NetProperties['Arrows'],
                edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
                linewidths=NetProperties['NodeBorderWidth'])
#nx.draw_networkx(grafo_copia,NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Num paths per edge', fontsize=10)
plt.show()
print('a')
```

```

# print(NumPathPorArco)
print('a')
# print(PathsQuePasanPorArco)
# print(DemandaPorArco)
print('a')

suma_arcos2 = {}
# Iterar sobre el diccionario y sumar los valores
for arco, valor in DemandaPorArco.items():
    # Ordenar el par
    arco_ordenado = tuple(sorted(arco))

    # Sumar el valor al diccionario
    suma_arcos2[arco_ordenado] = suma_arcos2.get(arco_ordenado, 0) + valor

# Lista de arcos con el número de caminos
# for arco, suma in suma_arcos2.items():
#     # print(f"{arco}: {suma}")
print('a')
MaxFlujoA = max([DemandaPorArco[(i, j)] for (i, j) in DemandaPorArco])
Link_scale = 10 # probar
Sizes = [Link_scale * (DemandaPorArco[(i, j)] / MaxFlujoA) for (i, j) in DemandaPorArco]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
                    wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(GAmp1, edgelist=Arcos2, pos=NodosPosiciones, node_size=8,
with_labels=NetProperties['Labels'],
                font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
                node_shape=NetProperties['NodeShape'],
                width=Sizes, arrows=NetProperties['Arrows'],
                edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
                linewidths=NetProperties['NodeBorderWidth'])
# nx.draw_networkx(grafo_copia, NodosPosiciones, with_labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Flujo de personas por arco', fontsize=10)
plt.show()
print('a')

# Inicializar el diccionario para almacenar el número de personas que pasan
por cada nodo
personas_por_nodo = {}

# Iterar sobre la lista de arcos y sumar las cantidades de personas a cada
nodo
for (origen, destino), personas in DemandaPorArco.items():
    if origen not in personas_por_nodo:
        personas_por_nodo[origen] = 0
    if destino not in personas_por_nodo:

```

```

    personas_por_nodo[destino] = 0

    personas_por_nodo[origen] += personas
    personas_por_nodo[destino] += personas

# Imprimir el resultado
#for nodo, total_personas in personas_por_nodo.items():
    #print(f"Nodo {nodo}: {total_personas} personas")
print('a')
print('a')

MaxFlujoN=max([personas_por_nodo[i] for i in personas_por_nodo])
Scale_centrality=100
Sizes=[Scale_centrality*(personas_por_nodo[i]/MaxFlujoN) for i in
personas_por_nodo]
Transparency=[(personas_por_nodo[i]/MaxFlujoN) for i in personas_por_nodo]
fig, ax = plt.subplots(nrows=1, ncols=1)
plt.subplots_adjust(top=0.95, bottom=0.01, right=1, left=0, hspace=0,
wspace=0)
ax.axis('equal')
ax.axis("off")
nx.draw_networkx(G, pos=NodosPosiciones, node_size=Sizes,
with_labels=NetProperties['Labels'],
                font_size=NetProperties['FontSize'],
node_color=NetProperties['NodeColor'],
                node_shape=NetProperties['NodeShape'],
                width=NetProperties['LinkWidth'],
arrows=NetProperties['Arrows'],
                edge_color=NetProperties['LinksColor'],
edgecolors=NetProperties['NodeBorderColor'],
                linewidths=NetProperties['NodeBorderWidth'],
alpha=Transparency)
#nx.draw_networkx(grafo_copia,NodosPosiciones, with_Labels=False,
node_size=Sizes, font_size=6, arrowstyle='-')
plt.title('Flujo de personas por Nodo', fontsize=10)
plt.show()

```