Proyecto Fin de Máster
Ingeniería Industrial

Análisis de Supervivencia en Infraestructuras Críticas

Autor: Pablo Martínez-Galán Fernández
Tutor: Juan Francisco Gómez Fernández

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

Sevilla, 2017

Proyecto Fin de Máster
Ingeniería Industrial

Análisis de Supervivencia en Infraestructuras Críticas

Autor:

Pablo Martínez-Galán Fernández

Tutor:

Juan Francisco Gómez Fernández

Profesor titular

Dep. Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Proyecto Fin de Carrera: Análisis de Supervivencia en Infraestructuras Críticas

Autor: Pablo Martínez-Galán Fernández

Tutor: Juan Francisco Gómez Fernández

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal

A mi familia

A mi tutor Juan F. Gómez

Agradecimientos

Agradecimiento muy especial a mi tutor del Trabajo de Fin de Máster, el doctor D. Juan Fco. Gómez Fernández, por su valiosa ayuda, dedicación y orientación, que ha hecho posible la finalización del presente trabajo y ha despertado mi inquietud por la investigación.

También me gustaría expresar mi agradecimiento a mi familia, que me ha apoyado en todo momento durante los años de estudio.

Pablo Martínez-Galán Fernández

Sevilla, 2017

Resumen

La dependencia del ser humano en el correcto funcionamiento de las redes de servicios ha ido incrementando con el tiempo. A la vez, también ha incrementado el riesgo ante fallos en las redes, principalmente debido al aumento de la complejidad y a las dependiencias presentes en las mismas. A raíz de este hecho surge la necesidad de aumentar la resiliencia de las redes, donde los costes de los incidentes se propagan de manera drástica a lo largo de la red, afectando a numerosos clientes e impactando tanto en los resultados como en la imagen de la empresa.

Combinando la fiabilidad con el *Input-Outup Inoperability Model* (IIM) y con una técnica novedosa como es el *Path Diversity*, se logra un modelo aplicable a todo tipo de redes y con unos resultados prometedores, confirmándolo como una valiosa herramienta de cara a la toma de decisiones de una organización. El modelo nos permite de manera rápida y eficiente determinar a aquellos puntos de la red con mayor riesgo, pudiendo así realizar un mantenimiento más eficiente sobre dichos puntos.

Abstract

Our society-increased dependence on utilities performance fosters the need for this infrastructures resilience. Especially in networks utilities where costs of incidents can enormously propagate throughout the network, affecting numerous customers and impacting business results, image and reputation.

Matching reliability, the Input-Output Inoperability Model (IIM) and the Path Diversity, we have achieved a model applicable to all networks with promising results, confirming it as a valuable tool for decision making of an organization. The model provides the highest risk points of the network which allows a more efficient maintenance on these points.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICAS

1 RESILIENCIA Y SUPERVIVENCIA DE REDES

Este capítulo introducirá los conceptos de resiliencia y de supervivencia, conceptos claves sobre los que se fundamentará el documento. Aunque las redes existen desde hace siglos, la protección y la supervivencia de las mismas no ha sido objeto de estudio hasta las últimas décadas. El desarrollo tecnológico ha provocado un severo aumento de la complejidad, la dependencia y la interconexión de las redes, lo que las ha vuelto más vulnerables, de aquí el interés que ha surgido por ellas. Catástrofes nacionales e internacionales durante los últimos años han puesto en jaque a los gobiernos y a las grandes empresas que se han dado cuenta de la importancia del análisis de las redes y de su entorno. Las políticas de los gobiernos y las empresas han ido evolucionando desde la protección de las redes hacia la resiliencia de las mismas, por lo que el objetivo actual se basa no sólo en protegerlas, sino en conseguir un funcionamiento contínuo de las mismas ante fallos tanto internos como debidos a su entorno.

1.1 Introducción

La dependencia del ser humano en el correcto funcionamiento de las redes ha ido incrementando con el tiempo, desde las antiguas civilizaciones como los romanos con sus carreteras o sus acuedutcos hasta los gobiernos actuales con sus plantas de potencia y sus redes eléctricas. A la vez que esta dependencia ha ido creciendo lo ha hecho el riesgo de fallos en las redes. Los últimos años nos dejan verios ejemplos de este hecho.

En 1998 el fallo del satélite *Galaxy IV* provocó el fallo de cerca del 90% de los busca en Estados Unidos, el fallo de multitud de servicios de los bancos como las tarjetas de crédito e incluso afectó al servicio médico del país debido al fallo en la comunicación con las ambulancias y los hospitales [25].

A comienzos de 2001 un fallo en una planta de potencia en California afectó a la producción de gas natural y petróleo, a las refinerías, y al transporte de las mismas tanto dentro de California como en los estados vecinos [25]. Este fallo también provocó la caída de industrias importantes que perdieron billones de dólares a causa de no poder producir.

Más de 15 años después continúan ocurriendo fallos que desencadenan en consecuencias catastróficas, como por ejemplo el ciberataque del pasado mes de Mayo en el que el ransomware (virus con código malicioso que bloquea los equipos de manera remota y solicita el pago de una cantidad de dinero en bitcoins para recuperarlos.) llamado *Wanna Cry* secuestraba los datos de los ordenadores y pedía un rescate de 300 dólares por dispositivo para liberar el sistema [35]. España fue el tercer país más afectado del mundo tras EEUU y Reino Unido. Los principales afectados de este ataque fueron los equipos de la sede de telefónica de Madrid, la red de hospitales británicos o el ministerio del Interior ruso. El masivo ataque provocó mas de 200.000 afectado en 150 países más otros cientos de miles en Asia unos días después. En la Fig. 1 se puede ver cómo quedó el mapa de afectados.

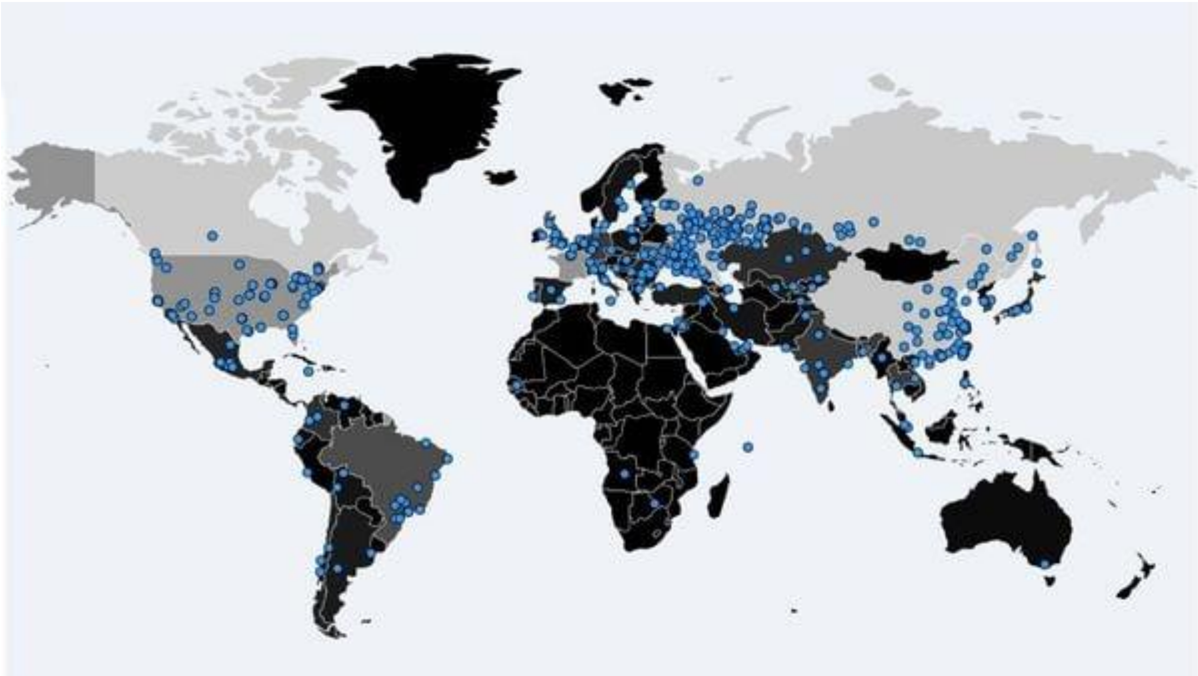


Figura 1 Países afectados por el ciberataque.

Este tipo de ataques producen una serie de consecuencias en cascada que pueden llegar a afectar a la estructura completa de un país, de ahí que sea imprescindible una red en la que no se produzcan fallos en cascada cuando se cae un punto de la red. Hay que intentar fortalecer los puntos más débiles de la red de manera que se minimice el riesgo de fallo en la red.

Una vez que se ha visto la fuerte dependencia del ser humano con el correcto funcionamiento de las redes, y el hecho de que las redes se están volviendo cada vez más complejas, existe la necesidad diseñar redes que sean capaces de resistir e incluso evitar los fallos de la red. Esto se consigue mediante la resiliencia de redes, concretamente mediante la supervivencia, una de las disciplinas de la resiliencia [21] que se define como la capacidad de un sistema para cumplir sus objetivos en presencia de amenazas como ataques o desastres naturales a gran escala.

El diseño de la supervivencia de redes se ha vuelto una tarea compleja puesto que la mayoría de los modelos desarrollados hasta el momento son modelos analíticos de simulación que conllevan un gran coste y tiempo asociados. Esta complejidad se debe principalmente a las dependencias existentes dentro de las redes, que son las principales culpables de que se produzcan los fallos en cascada.

El objetivo del trabajo es encontrar un modelo matemáticamente sencillo que permita localizar los puntos más débiles de red y por tanto aquellos puntos que más se deberían reforzar, de manera que ayude en la toma de decisiones de una organización.
La misión de reforzar estos puntos es evitar todo tipo de amenazas que se presenten, y en el caso de que ocurran fallos, que la red se reestructure de tal manera que pueda continuar proporcionando el servicio al que estaba destinada. Para asegurar este funcionamiento de las redes es fundamental centrarse en los términos de fiabilidad, resiliencia y supervivencia de las redes.

1.2 Dependencias en las redes

Las dependencias en las redes normalmente son complejas y nada obvias. Pueden originar fallos en cascada que afecten a diferentes puntos, siendo un potencial peligro para múltiples organizaciones, sectores o países. Hoy en día se usan cada vez más los modelos de dependencia para realizar los planes de emergencia, con el objetivo de prepararse mejor ante posibles fallos en cascada. Es necesario identificar bien las dependencias de cara a generar un buen modelo. Así se podrá identificar qué dependencias son las más críticas y adoptar una mayor seguridad o mayor control para reducir el riesgo global de la red.

Se define dependencia como "Una conexión entre dos infraestructuras a través de la cual el estado de una infraestructura influencia o está relacionado con el estado de la segunda". En la Fig. 2 se muestra el concepto de dependencia [24].

Bajo un correcto funcionamiento la planta de generación de electricidad requiere gas natural y petróleo para funcionar, carreteras, infraestructuras ferroviárias, tuberías para proporcionar el combustible a los generadores, conductos para transportar aire, agua para refrigeración, servicios de bancos y otros servicios financieros y sistemas de telecomunicaciones para el control. En el caso de que falle uno de estos sistemas que alimentan la planta de potencia, ésta tendrá varias dependencias críticas frente a las que tendrá que protegerse. Para este ejemplo la planta de potencia sería la infraestructura soportada mientras que el resto serían las infraestructuras de apoyo.

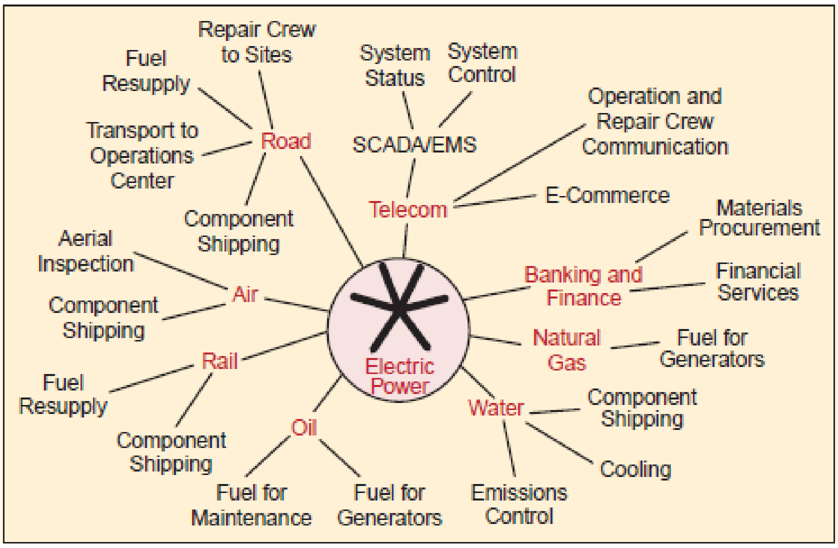


Figura 2 Ejemplo de dependencias de una planta de generación de potencia.

Además del término de dependencia, cuando hablamos de una red con múltiples infraestructuras tenemos que tener en cuenta el término *Interdependencia* que sería una relación bidireccional que existe entre dos infraestructuras a través de la cual el estado de una de ellas puede influenciar en el estado de la segunda y viceversa [25]. En otras palabras, dos infraestructuras son interdependientes cuando cuada una es dependiente de la otra. El problema de la interdependencia es que aumenta drásticamente la complejidad de las redes. Estas interdependencias se pueden describir en términos de varias dimensiones. Generalmente los autores suelen incluir entre 4 y 7, por ejemplo en [24] dividen en seis dimensiones que se muestran en la Fig. 3.

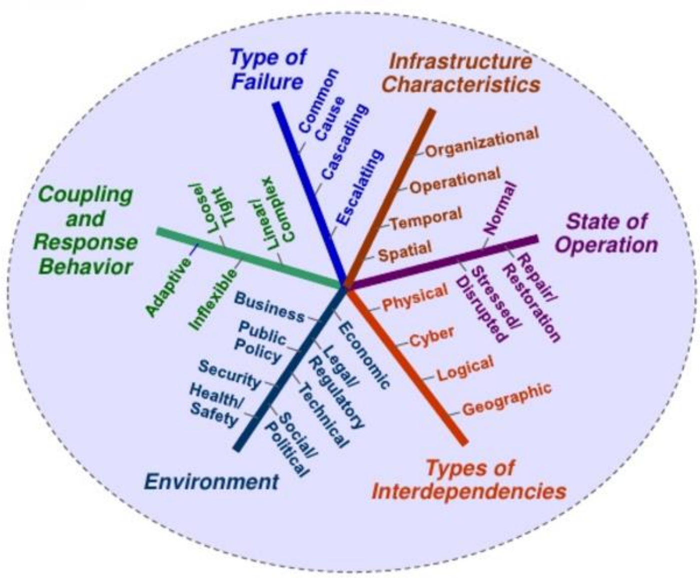


Figura 3 Dimensiones para describir las interdependencias entre infraestructuras.

La primera clasificación que se puede realizar es la del tipo de interdependecia, que puede ser física, cibernética, geográfica o lógica. Esta clasificación varía en función de la fuente de información, llegando algunos autores hasta enumerar siete categorías diferentes. En segundo lugar, una interdependencia se puede caracterizar por la naturalez de la misma. Pueden ser técnicas, económicas, de negocios, sociales, políticas, legales, públicas o de seguridad y salud. Esta naturaleza influye en el estado normal de operación, en el estado de emergencia durante fallos o períodos de alto estrés y en períodos de reparación o recuperación.

El grado en el que las infraestructuras están conectadas influye en gran medida en sus características operativas. Algunos de los enlaces son débiles, y por tanto flexibles, mientras que otros son fuertes, dejando poca o ninguna flexibilidad para que el sistema responda ante cambios inesperados o fallos en cascada desde otro sistema.

Según las características de las infraestructuras, las interdependencias pueden ser espaciales, temporales, operacionales u organizacionales y pueden afectar a su capacidad para adaptarse a los cambios en el sistema.

Por último estas interdependencias y las topologías de la infraestructura pueden crear interacciones y mecanismos de retroalimentación que a menudo conducen a comportamientos no intencionales y consecuencias negativas durante los fallos.

Para hacer frente a las fuertes dependencias que existen en las redes se requiere de la supervivencia de redes y del diseño de redes resilientes que permitan mantener un funcionamiento contínuo de la red y que un fallo singular no provoque una reacción en cadena que acabe con la red.

1.3 Importancia de la resiliencia

Las dependencias aumentan en gran medida la probabilidad de fallos en cascada en las redes. Los bucles de retroalimentación y las topologías complejas creadas por éstas pueden iniciar y propagar fallos de tantas maneras que son muy difíciles de prever. Según [24] se pueden clasificar los fallos relacionados con las interdependencias como fallos en cascada, fallos escalados o fallos con causa común.
Las estrategias de las distintas organizaciones han ido siguiendo una clara dirección hacia la resiliencia de las redes. La principal razón es que las redes se encuentran en un entorno con continuos cambios y fenómenos aleatorios que amenazan constantemente a las mismas, siendo imposible prevenirlos o preparase para evitar todos estos eventos, los cuales en su mayoría son desconocidos.

En la Fig. 4 [21] se pueden ver las distintas disciplinas de la resiliencia, divididas en dos grandes bloques. En primer lugar se ecnuentra la tolerancia a los desafíos, que incluye las disciplinas que se ocupan del diseño y la ingeniería de los sistemas que son capaces de prover un servicio continuado ante los eventos no deseados. El segundo bloque está formado por las disciplinas de la confiabilidad, que describen las propiedades medibles de los sistemas resilientes. La relación entre estos dos bloques es la robustez, que en el contexto, es la confiabilidad de un sistema cuando es desafiado.

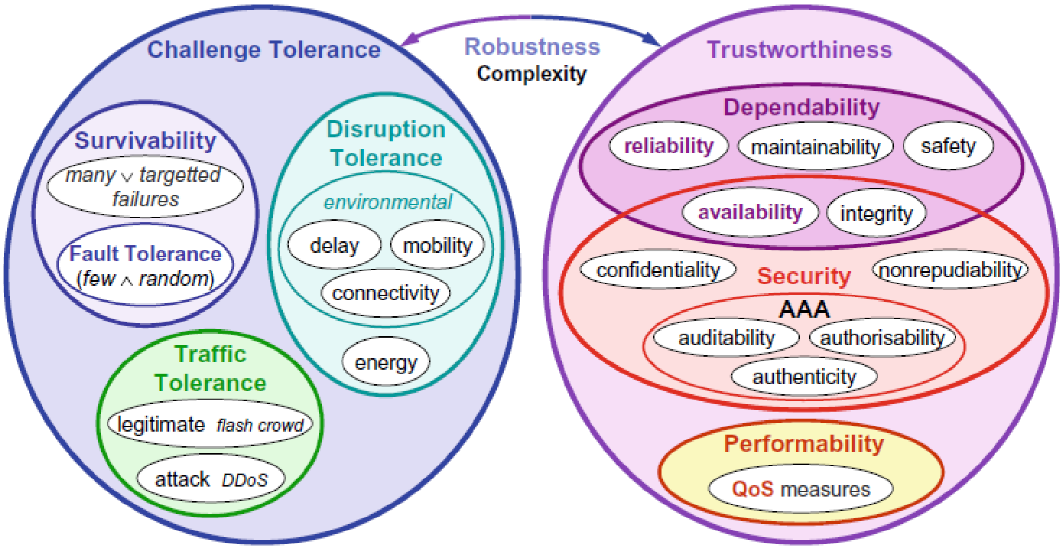


Figura 4 Disciplinas de la resiliencia

El concepto de resiliencia se ha convertido en en una referencia en muchas áreas de conocimiento, no sólo en la ingeniería. En la mayoría de los casos es un objetivo global que se trata desde una visión estratégica.

Se define resiliencia como capacidad inherente a cualquier sistema de proporcionar y mantener un nivel de servicio aceptable frente a fallos de cualquier tipo que perturben el estado normal de operación [26]. Autores como [28] definen la resiliencia de un sistema en función de las siguientes capacidades o características:

- Capacidad de absorción: Se refiere al grado con el cual el sistema puede absorber los impactos de las perturbaciones del sistema y minimizar las consecuencias con un pequeño esfuerzo. Los sistemas que presentan buena robustez y fiabilidad suelen ser sistemas resilientes.

- Capacidad adaptativa: Es la abilidad de un sistema de adaptarse a situaciones indeseables realizando algunos cambios. La capacidad adaptativa de un sistema crece con su habilidad para anticipar los fallos, reconocerlos y reorganizarse una vez que han ocurrido.

- Capacidad restaurativa: Un sistema resiliente se caracteriza por la rapidez con la que el sistema vuelve a su situación normal y su nivel de fiabilidad. Esta capacidad debe de ser evaluada para el nivel de servicio que se requiera.

Aunque según [28] la anticipación, que trata de evitar o predecir los fallos, está incluida dentro de la capacidad de adaptación, otros autores también incluyen como una capacidad aparte. En el presente trabajo se pretende dotar a un sistema de manera directa de las capacidades de absorción, adaptación y anticipación, puesto que se trata de minimizar las consecuencias de los fallos y de conseguir que las redes se adapten mientras se corrigen los fallos y de evitar que los fallos ocurran mediante la mejora del mantenimiento en los puntos con mayor riesgo de la red. También de manera indirecta se pretende mejorar la capacidad restaurativa. Mediante el análisis de riesgo que se lleva a cabo se puede saber cuales son los puntos más críticos tras un fallos y así corregir los errores de manera más rápida permitiendo la vuelta al estado normal de funcionamiento en un período menor de tiempo.

Vivimos en momentos de rápido cambio y debemos aprender a lidiar con él. Por eso la resiliencia es uno de los objetivos clave. En [29] proponen la resiliencia como la principal característica de las redes de cara a evitar los fallos de la red en cascada.
Los modelos basados en la resiliencia de las redes se están usando cada vez más en distintos tipos de organizaciones. Estos modelos les permiten afrontar la protección de las redes en un entorno que cada vez se hace más complejo y desconocido, superando así las limitaciones que tenían los modelos antiguos basados en los fallos históricos, los cuales no eran efectivos contra fallos o amenazas desconocidas.

1.4 Importancia de la supervivencia

La supervivencia de redes es una de las disciplinas que hacen a una red resiliente, y se ha ido haciendo importante a lo largo de los años. Los primeros artículos que aparecieron sobre la resiliencia de redes datan de los años 1950. Estas investigaciones se basaban en la tolerancia a los fallos, la primera de las disciplinas de la resiliencia. Las técnicas utilizadas para hacer a una red tolerante a los fallos se basaban en la redundancia, pero más adelante se dieron cuenta que no era suficiente, puesto que ante fallos a mayor escala se podía perder esta redundancia. Con la aparición de las nuevas tecnologías y las grandes dependencias dentro de las redes surgió la necesidad de investigar nuevas técnicas de diseño que evitara los fallos en cascada.
En los años 70 comienzan las primeras investigaciones sobre la supervivencia de redes, una técnica que tenía en cuenta tanto la diversidad en las redes como la redundancia requerida por la tolerancia a los fallos. Desde entonces se han llevado a cabo multitud de investigaciones sobre la supervivencia de redes y se han desarrollado numerosos modelos

de diseño de redes, algunos de carácter más general y otros sobre tipos de redes concretas como las redes de telecomunicaciones.

Después de revisar gran parte de la literatura se llega a la conclusión que la mayoría de las investigaciones y artículos están centrados en el diseño de nuevas redes, pero existen pocos modelos para evaluar la supervivencia de los puntos de una red ya existente. Sin embargo, las bases del diseño de la supervivencia de redes aportan una información muy valiosa a la hora de construir el modelo que se pretende.

Teniendo en mente que se quiere un modelo sencillo, que sea práctico y que no conlleve un elevado tiempo ni coste, se buscará en la literatura revisada las características de los modelos que cumplen algunos de estos requisitos. La finalidad del modelo es cuantificar el impacto que un nodo tiene sobre la supervivencia de otro. En la Fig. 5 se muestra el camino que se ha seguido para llegar a las bases del modelo que se presenta en el capítulo posterior.

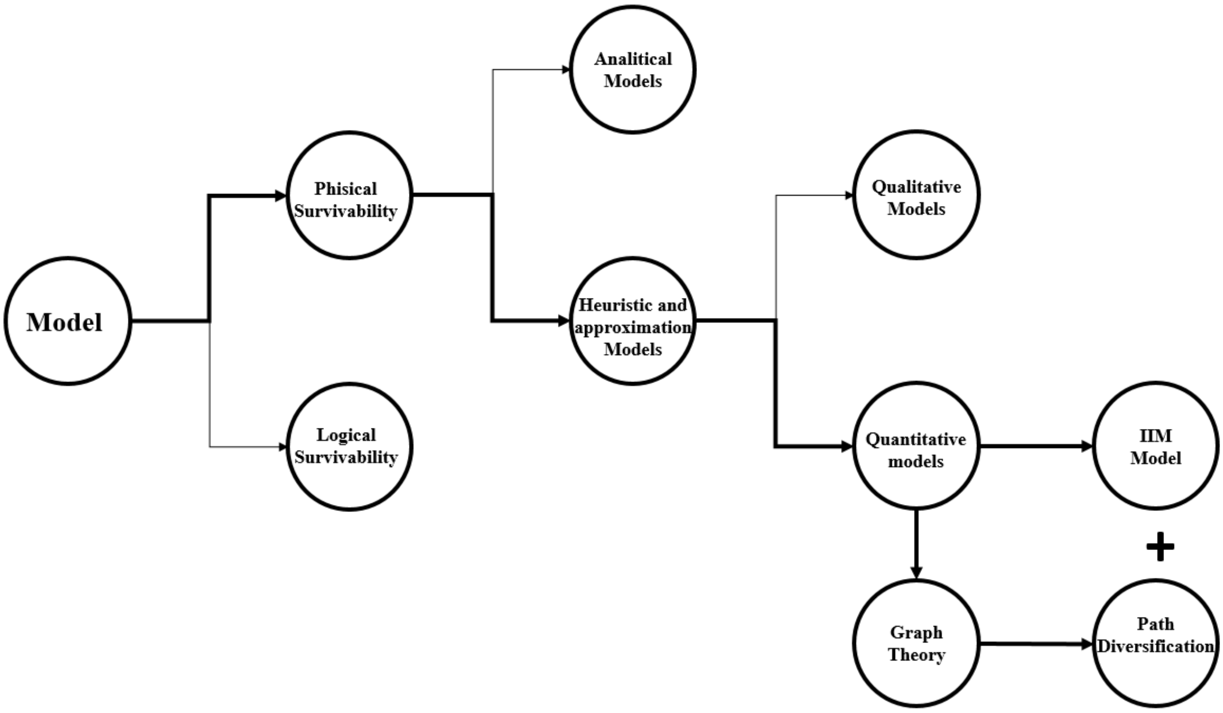


Figura 5 Diferentes modelos en la literatura de supervivencia de redes

Modelos de supervivencia física vs modelos de supervivencia lógica.

- Se pueden encontrar diversas revisiones de la documentación relacionada con la supervivencia de redes como en [3], donde se ofrece una clasificación muy interesante sobre los modelos de supervivencia de redes. Divide la documentación revisada en modelos de supervivencia física y modelos de supervivencia lógica. Los primeros se ocupan del diseño topológico de las redes. Dado un conjunto de nodos, de enlaces posibles, los costes de los enlaces y el número máximo permitido de fallos de enlaces y nodos entre cada par de nodos, el objetivo es diseñar una red lo más rentable posible teniendo en cuenta que cuanto mayor sea la el número de conexiones entre los nodos, mayor será la supervivencia, pero también lo será el coste. Estos modelos sólo consideran el coste asociado a la topología de la red. Los modelos de supervivencia lógica abordan el problema de enrutamiento y enlace. Si queremos aumentar la supervivencia de una red, debe de haber más de una ruta por la que se puedan distribuir las capacidades. De esta manera, si una ruta falla, la capacidad se puede desviar por una alternativa. Es importante tener esto en cuenta a la hora de diseñar la red puesto que se tienen que sobredimensionar las capacidades de los nodos y enlaces de manera que ante un fallo las capacidades se puedan reestructurar.

Parece un buen punto de partida el escoger que tipo de modelo se quiere construir. Puesto que el modelo que se quiere construir se va a basar principalmente en la topología de la red, parece lógico que escojamos un modelo físico. En el presente trabajo se ha supuesto que todos los nodos y enlaces tienen capacidad ilimitada. No obstante, se ha abierto una línea de investigación que tiene en cuenta ambos tipos de modelos, basándose además de en la topología de la red, en la capacidad que pueden asumir todos los enlaces y nodos.

En [8] se presentan varios tipos de modelos físicos que clasifican en dos tipos. Unos que se basan en la probabilidad de fallo de los enlaces y otros modelos condicionales que proporcionan parámetros de la red tras determinados fallos suponiendo los estados sucesivos por los que va a pasar la red. Por los requisitos comentados anteriormente el modelo a construir será más similar al primer tipo.

Modelos analíticos vs modelos heurísticos y de aproximación.

- En [9] se presenta otra revisión de la documentación y se llega a una conclusión que marca la segunda característica del modelo. En los modelos analíticos el tiempo requerido y el número de restricciones crecen de manera astronómica con el tamaño de la red. Sin embargo, los modelos heurísticos y de aproximación han resultado ser prometedores, con buenos resultados y bajos tiempos de computación. Por ejemplo, los artículos [12,13] hacen incapié en el buen funcionamiento de los modelos heurísticos.

Esto nos conduce a la idea de desarrollar un modelo heurístico que nos permita llegar a una buena solución de manera rápida. Como ejemplo de estos modelos, en [10] se presenta la modificación de un modelo analítico que produce una gran reducción en el tiempo requerido sin perder prácticamente exactitud. En [11] se presenta otra técnica de simplificación de los modelos analíticos: dividir para conquistar. Mediante la descomposición del problema en problemas más pequeños consiguen llegar a una buena solución de una manera más rápida y sencilla.

Modelos cualitativos vs modelos cuantitativos

- Una vez que hemos escogido un modelo físico y heurístico se nos presenta una nueva elección tras la división que se hace en [6] en modelos cuantitativos y modelos cualitativos. En modelos como [23] utilizan el juicio de expertos en lugar de medidas cuantitativas. Puesto que se huye de la subjetividad de los modelos cualitativos, se quiere llegar a una medida cuantitativa que nos permita medir el riesgo de los puntos de la red sin subjetividad alguna. Por tanto se escoge un modelo cuantitativo.

En el artículo [6] también se cuestionan que parámetros son los mejores para medir la supervivencia de las redes. Al no existir una medida directa de la supervivencia hay que encontrar un parámetro que la cuantifique de manera efectiva.

Path Diversity.

- Para dar solución al problema de la medida de la supervivencia, en [5] se introduce la diversidad de caminos, un concepto muy relacionado con la misma. Para dar validez a este nuevo conecepto, en [4] se realiza un estudio sobre los parámetros más efectivos para medir la supervivencia de una red. Se demuestra que al ordenar distintas redes con respecto a su supervivencia y con respecto a su diversidad, las clasificaciones son idénticas, por lo que se establece la diversidad como una buena medida de supervivencia.

Teoría de Grafos.

- Para calcular la diversidad en una red, los artículos [14,15] proporcionan la idea de basarse en la teoría de grafos y desarrollan multitud de algoritmos para encontrar caminos independientes. El modelo que se ha desarrollado en este trabajo, que permite calcular el número de caminos independientes entre cada par de nodos, en parte se basan en los algoritmos de dichos artículos.

IIM Model.

- Una vez que se tiene la medida de la supervivencia se quiere calcular el impacto que un nodo tiene sobre otro con el fin de cuantificar la supervivencia de un nodo en función del estado de los demás. El artículo [16] relaciona la supervivencia de redes con la disponibilidad de los servicios y en [23], que correponde con el "Input-Output Inoperability Model" (IIM), se calcula el impacto de un nodo sobre otro de acuerdo la degradación que produce en un nodo una reducción del servicio de otro. Por tanto, relacionando estas dos ideas se propone un modelo en el que el impacto se calcule en función de la reducción de diversidad que un nodo provoca sobre otro. Vamos a basar nuestro modelo en los impactos que tienen las degradaciones de los nodos en la red, por tanto la fiabilidad es un parámetro clave para medir esta degradación.

Finalmente, con los requisitos propuestos, se ha llegado a un modelo basado en topología, cuantitativo para evitar la subjetividad, basado en una técnica novedosa y bien relacionada con la supervivencia como es el *Path Diversity*, en la teoría de grafos y en el IIM.

1.5 Fiabilidad

En el presente trabajo como se ha comentado anteriormente, se crea un modelo para medir el riesgo de los nodos en cada instante de tiempo y saber donde hay que atacar en el preciso instante. La base para calcular este riesgo además de la resiliencia y la supervivencia va a ser la fiabilidad de los distintos componentes de la red. Se va a modelar la degradación de los distintos puntos de la red como la probabilidad acumulada de que fallen, es decir, el complementario de la fiabilidad.

De acuerdo con la comisión electrotécnica internacional [30], la fiabilidad se define como la habilidad de un item de proporcionar una determinada función requerida bajo unas condiciones dadas en un intervalo de tiempo determinado. Esta fiabilidad se puede expresar como la probabilidad de llevar a cabo la función requerida dentro de un nivel de actuación determinado.

Como se puede ver en la Fig. 6 extraída de [31], la fiabilidad se encuentra un nivel por debajo de la disponibilidad, que al fin y al cabo, es lo que se quiere conseguir. Así mismo, la disponibilidad esta un nivel por debajo de lo que se denomina confiabilidad, que a su vez, está estrechamente relacionada con la supervivencia y con la calidad tangible.

Como con la confiabilidad hay que tener en cuenta las condiciones operacionales del equipo, la fiabilidad debe estudiarse con cuidado pues no solo dependie de las condiciones operacionales, sino también de las condiciones del entorno, la combinación de condiciones de operación y condiciones del entorno es lo que de denomina contexto operacional del equipo.

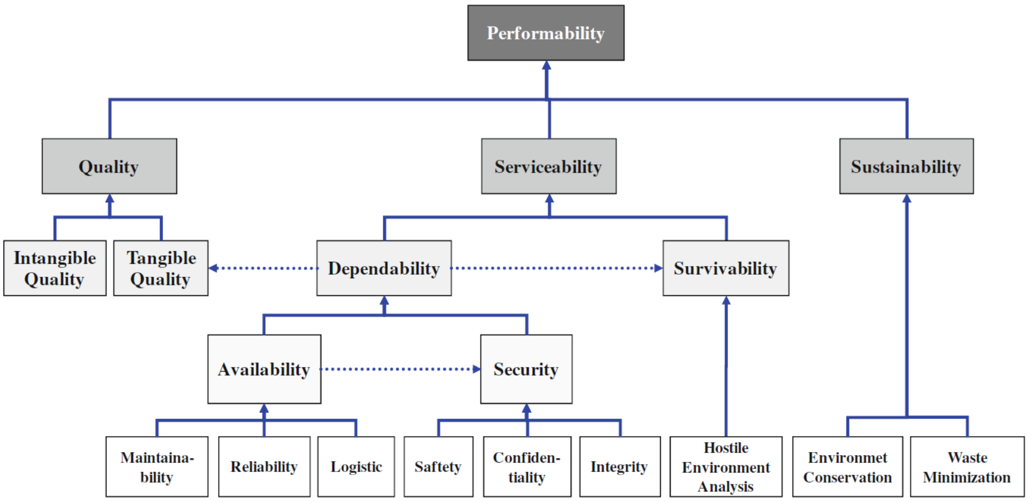


Figura 6 Términos de actuación en mantenimiento

A menudo ocurren fallos en los equipos producidos por causas atribuibles a condiciones de funcionamiento no óptimas (limpieza, fijación, temperatura, etc.), a cambios operacionales (configuraciones, mantenimiento preventivo, manejo indebido, etc.) y a defectos en los equipos (imperfección del diseño, errores de implementación, calidad de los materiales, etc.). Estas causas se pueden clasificar de la siguiente manera [32]:

• Físicos cuando la causa física es la razón por la cual el activo falló, la explicación técnica de por qué las cosas se rompieron o fallaron.

• Humanos cuando se producen como resultado de errores humanos, que resulta en raíces físicas.

• Los latentes cuando son inducidos por deficiencias en los sistemas de gestión que permiten que los errores humanos continúen sin control (fallas en los sistemas y procedimientos).

En general, debido a la incertidumbre de la aparición de fallos, el estudio de los mismos se suele llevar a cabo con variables aleatorias caracterizadas por funciones de densidad de probabilidad que permiten obtener buenas aproximaciones.

Por lo tanto, para conocer el comportamiento de las variables aleatorias es crucial hacer buen uso de los datos históricos de fallos, basándose en la propia experiencia, o en las bases de datos aceptadas, así como los libros escritos sobre ellas, como EuReDatA, PERD u OREDA.

El mantenimiento centrado en la fiabilidad (RCM) [36] es el método más extendido para estudiar el comportamiento de los equipos ante fallos que afectan a sus funciones requeridas en unas condiciones de operación determinadas, analizando las causas, cuantificando los riesgos y evaluando las medidas correctivas para detectar, evitar o prevenir los fallos.

La implementación del RCM lleva a equipos más seguros y fiables, a reducciones de costes, a una mejora en la calidad del producto, y a un mayor cumplimiento de las normas de seguridad y medio ambiente. Además, también está asociado a beneficios humanos, como puede ser la mejora en la relación entre distintas áreas de una empresa. Generalmente está asociado a un mejor entendimiento entre mantenimiento y operaciones.

Volviendo a la fiabilidad, para hacer frente a la incertidumbre de la aparición de fallos, nos interesa establecer una relación matemática entre el tiempo de uso de un equipo y la frecuencia con la que aparecen dichos fallos, permitiendo así un mejor mantenimiento. Para definir esta relación se utiliza la fiabilidad, probabilidad de que el equipo cumpla la función o funciones requerida en el instante t, y se define como R(t).

La fiabilidad está estrechamente relacionada con la función de probabilidad de fallo como,

$$F(t) = 1 - R(t) \tag{1}$$

Es decir, que la probabilidad de fallo acumulada es el opuesto a la fiabilidad. Ambas funciones están caracterizadas por la tasa de fallo (λ) la cual se define como el número de fallos en un tiempo determinado. Puesto que la tasa de fallo es variable en el tiempo, su representación típica tiene la forma de una bañera debido a los distintos comportamientos que presenta un equipo a lo largo de su vida útil. Un ejemplo se puede ver en la Fig. 7.

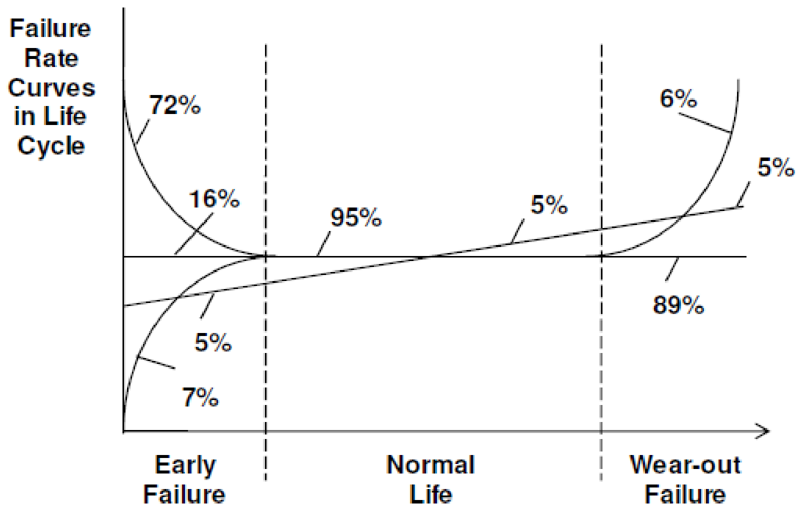


Figura 7 Variación típica de la tasa de fallo con el tiempo

En la curva se pueden diferenciar tres etapas:

1. Zona de fallos iniciales: La primera etapa se caracteriza por presentar una alta tasa de fallos que va descendiendo rápidamente conforme pasa el tiempo. Generalmente estos fallos se asocian a instalaciones incorrectas, equipos defectuosos, errores de diseño, etc.

2. Zona de funcionamiento normal: La segunda etapa presenta una tasa de fallo menor y prácticamente constant. Esto se debe a que los equipos ya no fallan por causas internas, sino por causas externas que no se pueden prever como un accidente o una mal uso del equipo.
3. Zona de envejecimiento: Por ultimo la tercera etapa está formada por una tasa de fallo que crece drásticamente. Los fallos se producen por el deterioro de los equipos al llegar al final de su vida útil.

Al analizar los datos históricos de una planta y calcular las tasas de fallo se puede saber en que parte de la curva estamos. Aunque realmente existen varios tipos diferentes de curva de bañera dependiendo del tipo de componente del que se trate.

Según la variación de la tasa de fallo se utilizan unas distribuciones u otras en el terreno de la fiabilidad. Las distribuciones más típicas son la distribución exponencial y la distribución de Weibull.

1.5.1 La distribución exponencial

La distribución exponencial juega un papel clave en la teoría y la práctica de la fiabilidad, puesto que describe con una buena aproximación las características de fallo de muchos de los equipos durante el funcionamiento [31].

Como se veía en la curva de bañera, en la zona normal de funcionamiento la tasa de fallo es constante, y por tanto esta distribución es correcta utilizarla en dicho período. Si queremos caracterizar los primeros o los últimos años de un equipo no es la distribución apropiada.

$$\lambda(t) = \lambda \quad (2)$$

Matemáticamente la función de densidad de probabilidad de fallo se puede escribir como:

$$f(t) = \lambda \cdot e^{-\lambda \cdot t} \quad (3)$$

Integrando dicha función se obtiene la probabilidad de fallo acumulada F(t).

$$F(t) = 1 - e^{-\lambda \cdot t} \tag{4}$$

Y por tanto la fiabilidad tiene la siguiente expresión:

$$R(t) = 1 - F(t) = e^{-\lambda \cdot t} \tag{5}$$

La fiabilidad representa por tanto la probabilidad de que un equipo , el cual está caracterizado por una tasa de fallo constante, no falle durante un período de funcionamiento t. Por ejemplo, un equipo que presenta una tasa de fallo constante de 0.1 fallos/año, tiene una probabilidad de llegar al sexto año sin fallo de:

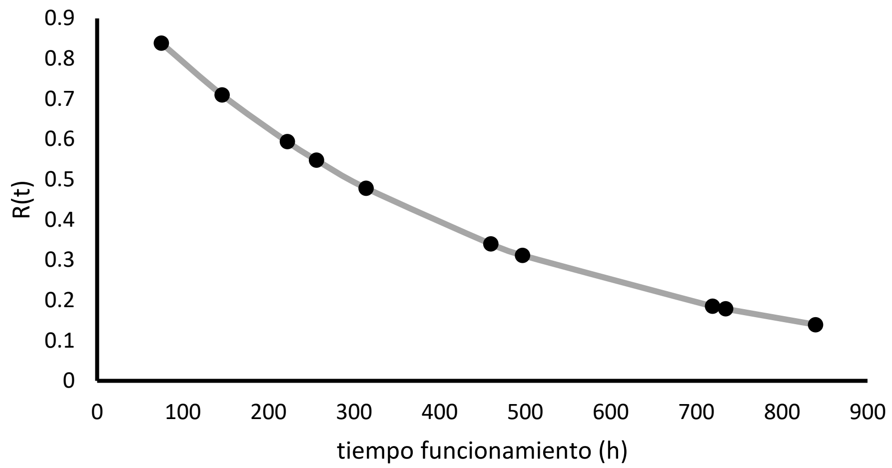
$$R(6) = e^{-0.1 \cdot 6} = 0.55 \tag{6}$$

Por tanto, el equipo tiene una probabilidad del 55% de llegar al sexton año sin fallar. Cuantos más datos históricos de fallos y más experiencia tengamos, más exacta sea la tasa de fallo y mejor serán las aproximaciones.

Puesto que la tasa de fallos es constante e indica el número de fallos por período de tiempo, es fácil calcular el tiempo medio de funcionamiento hasta el fallo (MTTF) como:

$$MTTF = \frac{1}{\lambda} \tag{7}$$

En la gráfica 1 se muestra la curva típica de la fiabilidad para los equipos con tasa de fallo constante.



Gráfica 1 Fiabilidad en equipo con distribución exponencial

1.5.2 Distribución de Weibull

Al contrario que la exponencial que solo se ajusta a aquellos equipos que presentan una tasa de fallo constante, la distribución de Weibull es mucho más flexible y permite con los tres parámetros que presenta, un ajuste correcto de toda clase de datos experimentales y operacionales [31]. Además, al no ser la tasa de fallo constante también se pueden modelar la zona de fallos iniciales y la zona de envejecimiento. En contrapartida requiere más información que la exponencial para un mejor ajuste.

La función de densidad de probabilidad de fallo ese en este caso:

$$f(t) = \left(\frac{\beta \cdot t^{\beta-1}}{\alpha^{\beta}}\right) \cdot e^{-(\frac{t}{\alpha})^{\beta}} \quad (8)$$

Que integrándola resulta en la siguiente función de probabilidad de fallo acumulada:

$$F(t) = 1 - e^{-(\frac{t}{\alpha})^{\beta}} \quad (9)$$

Que como se puede observar, cuando β=1, coincide con la distribución exponencial. Sabiendo la probabilidad de fallo acumulada, la función de fiabilidad resulta:

$$R(t) = e^{-(\frac{t}{\alpha})^{\beta}} \quad (10)$$

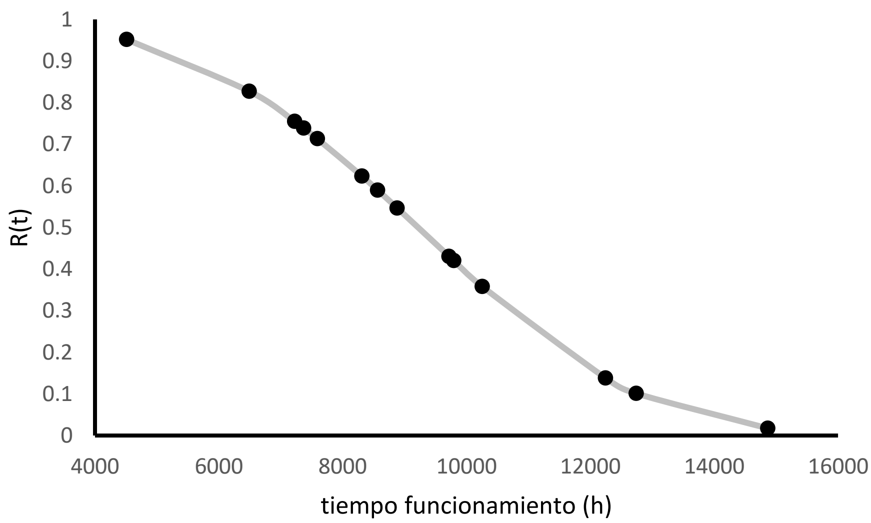
Y la tasa de fallo, que no es constante:

$$\lambda(t) = \frac{\beta}{\alpha} \cdot (\frac{t}{\alpha})^{\beta-1} \quad (11)$$

El conocimiento del parámetro β es muy valioso a la hora de diagnosticar el tipo de fallo. Si $\beta < 1$ la tasa de fallo decrece y por tanto el equipo se encuentra en el período de fallos iniciales. Si $\beta = 1$ la tasa de fallo es constant, independientemente del proceso y del tiempo. Si $\beta > 1$ la tasa de fallo crece y por tanto estamos en la parte de envejecimiento u obsolescencia.

Mientras que algunos equipos como los electrónicos muestran una larga fase de su vida con la tasa de fallo constant, otros equipos como los mecánicos, a causa del desgaste, no muestran tal aplanamiento en la curva, y por tanto deben de ser modelados con una Weibull en lugar de con una exponencial como en el primer caso.

En la gráfica 2 se muestra un ejemplo de la curva de fiabilidad para una distribución de Weibull.



Gráfica 2 Fiabilidad en equipo con distribución de Weibull

1.5.3 Fiabilidad de sistemas

El problema básico de la fiabilidad de sistemas consiste en el cálculo de la fiabilidad, a partir de las fiabilidades de los equipos que lo forman.

A continuación se muestran las configuraciones básicas más usuales que forman los equipos de un sistema.

1. Sistemas en serie.

En primer lugar se denomina un sistema en serie a un sistema en el que el fallo de un equipo provoca el fallo del sistema total, es decir, solo funciona si todos los equipos funcionan de manera correcta.

Para un número x de equipos en serie la fiabilidad del sistema vendrá dada por el producto de las fiabilidades de los equipos del sistema.

$$R(t) = \prod_{i=1}^{x} R_i(t) \tag{12}$$

Por tanto, la fiabilidad de un sistema que presenta los equipos en serie se puede aumentar reduciendo el número de equipos, aumentando la fiabilidad de los mismos, o ambas cosas.

2. Sistemas en paralelo

Al contrario que los sistemas en serie, en los sistemas en paralelo el fallo de un equipo no implica el fallo del sistema, en este caso tienen que fallar todos los equipos a la vez para que falle el sistema. En consecuencia, la fiabilidad de estos sistemas será bastante mayor que la de los sitemas en serie.

Puesto que el sistema falla si fallan todos los equipos la probabilidad de fallo del sistema será:

$$F(t) = \prod_{i=1}^{x} F_i(t) \tag{13}$$

Y por tanto la fiabilidad será:

$$R(t) = 1 - \prod_{i=1}^{x} F_i(t) = 1 - \prod_{i=1}^{x} (1 - R_i(t)) \qquad (14)$$

La característica que presentan los sitemas en paralelo se denomina redundancia, es decir, que existe más de un equipo para realizar la misma función. Se puede dividir en dos tipos.

1. Redundancia activa. Todos los elementos en paralelo están activos al mismo tiempo durante el funcionamiento del sistema.
2. Redundancia pasiva. Los equipos redundantes solo entran en funcionamiento cuando fallan los principales, hasta este momento han permanecido inactivos.

Recordemos que la redundancia hacía frente a la tolerancia a los fallos en los sistemas, pero que no hacía a una red totalmente resiliente puesto que los equipos redundantes por encontrarse en zonas cercanas pueden fallar a la vez debido a eventos catastróficos. Es por esto que se propone la diversidad como principal característica para hacer a una red resiliente.

Una vez que podemos cuantificar el impacto entre los nodos, se desarrolla en el siguiente punto un modelo basado en la diversidad de caminos, en la fiabilidad, en la teoría de grafos y en el IIM. Esto nos permite realizar una clasificación de la supervivencia de los nodos.

2 MODELO PROPUESTO

EN este segundo capítulo se va a desarrollar el modelo propuesto para localizar los puntos más débiles de una red en cada instante de tiempo, en función de la fiabilidad de cada punto de la red y de los fallos externos que puedan surgir. El objetivo es facilitar la toma de decisiones de una organización, de manera que se consuman los mínimos recursos posibles en reforzar aquellos puntos que en el momento en cuestión o bien más adelante, son grandes candidatos a provocar un fallo en la red. Las características del modelo permiten una rápida actualización de la red ante eventos indeseados, permitiendo así saber los puntos críticos en los siguientes instantes de tiempo y las consecuencias de no reparar dichos puntos.

2.1 Introducción

Como se ha comentado en el primer capítulo se va a presentar una aproximación heurística que se va a basar en la propagación de la degradación a lo largo de una red. Se propone un modelo construido sobre el Input-Output Inoperability Model (IIM), introducido en [23] como una evolución del modelo input-output del premio Nobel Leontief. El IIM se centra en la difusión de la inoperabilidad a través de la red, definiendo el término de inoperabilidad como una degradación parcial o total de la producción normal.

Aunque el IIM está orientado hacia fallos indeseados se quiere construir un modelo que no sólo tenga en cuenta estos fallos sino que también tenga en cuenta la fiabilidad de los componentes de la red. Esto se va a llevar a cabo midiendo la degradación como la probabilidad de fallo que presenta cada punto de la red a lo largo del tiempo. De esta manera los puntos de la red que estén más degradados serán más fácil que caigan ante fallos en cascada que los puntos de la red que no presenten degradación.

Frente a la multitud de modelos de diseño de redes que existen se propone un modelo que sirve para el diseño de nuevas redes y principalmente para medir la supervivencia en redes ya existentes. Las bases del modelo como ya se comentaba en el punto anterior son la teoría de grafos, la fiabilidad, la diversidad de caminos y el IIM.

2.2 Input-Output Inoperability model (IIM)

El IIM es el modelo más usado dentro del marco de las aproximaciones holísticas [25]. Con un alto nivel de aproximación, el IIM asume que cada punto de la red puede ser tratado como una unidad en la que el nivel de operabilidad depende de la disponibilidad de recursos que proporcionan los demás puntos de la red. Si un punto de la red falla, provoca una reducción de servicios en los puntos a los que suministra, lo cual se denomina una degradación del servicio. Estas degradaciones se pueden propagar a lo largo de la red causando un fallo en cascada con consecuencias graves para la misma.

Matemáticamente, el IIM describe este fenómeno de propagación de la degradación, basado en el nivel de inoperabilidad de cada punto de la red [23]. La inoperabilidad se corresponde con el porcentaje en el que se han reducido los servicios de un determinado punto. Tras una serie de operaciones matemáticas se llega a que la inoperabilidad de los puntos de la red causada por una perturbación puede ser evaluada como:

$$x(t+1) = A \cdot x(t) + c \qquad (15)$$

Donde x y c son vectores que respresentan la inoperabilidad de los puntos de la red y la reducción de la operabilidad inicial asociada a la perturbación respectivamente y A es una matriz de dimensiones $n\ x\ n$ llamada matriz de influencia. La matriz de influencia modela el efecto directo debido a las dependencias de primer orden entre los nodos, y los coeficientes de esta matriz a_{ij} representan la inoperabilidad causada en el nodo i cuando el nodo j está completamente inoperable.

Relacionado con la matriz de influencia, en [33] se introducen el índice de dependencia y la ganancia de influencia. El índice de dependencia se define como la suma de todos los coeficientes a_{ij} a lo largo de las filas,

$$\sum_{j=1}^{n} a_{ij} = \delta_i \qquad (16)$$

y mide la robustez de un nodo con respecto a la inoperabilidad de los demás nodos. Es decir, representa la máxima inoperabilidad del nodo i cuando el resto de nodos están completamente inoperables. En segundo lugar la ganancia de influencia se mide como la suma de los coeficientes a_{ij} a lo largo de las columnas de la matriz de influencia.

$$\sum_{i=1}^{n} a_{ij} = \rho_j \qquad (17)$$

Al contrario que el índice de dependencia, la ganancia de influencia mide el impacto que un nodo tiene sobre la red en términos de inoperabilidad. Un nodo con un alto valor de este índice indica que la inoperabilidad de este nodo daña notablemente a la red, mientras que un bajo valor de este índice indica que la inoperabilidad de dicho nodo no va a afectar prácticamente en el resto de nodos de la red.

Volviendo a la ecuación (15), podemos despejar la inoperabilidad correspondiente a una perturbación c como:

$$x = (I - A)^{-1} \cdot c = S \cdot c \qquad (18)$$

Donde la matriz S tiene en cuenta también las amplificaciones introducidas por las dependencias de segundo orden y superiores.
En [31] la matriz de influencia se calcula mediante el conocimiento aportado por expertos y técnicos. Estiman los coeficientes a_{ij} con la ayuda de numerosos expertos que evalúan el impacto que causan sobre su infraestructura una ausencia de los servicios proporcionados por el resto de infraestructuras. Para ello, se mandan cuestionarios específicos para cuantificar el impacto según unas tablas normalizadas. Además, el experto tiene que calificar la confianza sobre su evaluación usando una segunda tabla normalizada. Cuanto mayor sea la confianza menor será el rango de impacto y por tanto más exacta será la aproximación.
Este cuestionario se repite varias veces para distintas duraciones de la ausencia de servicios. Finalmente se introduce un coeficiente de corrección en función de la experiencia y de la trayectoria de cada experto.

A partir de la matriz de influencia, la matriz S se calcula matemáticamente como se puede ver en la ecuación (18). Con respecto al modelo, en un primer momento se trató de encontrar una matriz de influencia basada en propiedades de la red en lugar de en opiniones de expertos para introducirla luego en el IIM, pero se llegó a la conclusión de que la matriz de influencia es muy restrictiva y que sólo funciona con redes pequeñas. Además cuanto mayor es el tamaño de la red, más complicaciones se introducen. En la mayoría de los casos es necesario que la suma de las columnas de la matriz de influencia sea menor que 1 para que el IIM converja.

$$\sum_{i=1}^{n} a_{ij} < 1 \qquad (19)$$

Esta condición no es estrictamente necesaria cuando estamos tratando con redes pequeñas, pero para redes mayores es esencial para poder calcular la matriz S y que el IIM funcione.
Imaginemos una situación en la que dos nodos A y B dependen completamente de un tercero C. El impacto por tanto que tiene el nodo C sobre los nodos A y B sería de un 100% sobre cada uno, por lo que tendríamos ya una suma de 2 o más en la columna correspondiente al impacto de C sobre el resto de la red. Esta situación genera que no se pueda calcular la matriz S y por tanto que no se pueda aplicar el IIM. Por tanto, no se puede utilizar el IIM calculando la matriz de influencia de una manera objetiva, o al menos en la mayoría de los casos.
Puesto que se quiere realizar un modelo con la máxima aproximación posible, vamos a huir de la subjetividad de las opiniones de expertos que utiliza el IIM y vamos a caluclar una matriz S que represente realmente el impacto de un nodo sobre los demás de la red y que se calcule en función de un parámetro de la red que se verá a continuación. Se mantiene la estructura del modelo IIM pero se calcula directamente la matriz S como una matriz de impacto objetiva.

2.3 Diversidad de caminos

La segunda base teórica del modelo propuesto es la diversidad de caminos, introducida en [5]. Surge como nuevo concepto alrededor de los años 70 para complementar a la redundancia que hacía frente a la tolerancia de fallos. Cuando se vió que fallos como las catástrofes naturales anulaban la redundancia y esta resultaba inútil, en casos como ese fue cuando se empezó a investigar sobre la diversidad de caminos. Desde entonces se han diseñado multitud de algoritmos para calcular caminos independientes, K caminos independientes con longitud mínima, etc.

Con la mentalidad de incrementar la resiliencia se propone en [5] seleccionar caminos que no puedan caer ante un mismo fallo, es decir, que sean diversos. Además, en [4] se realiza un estudio sobre los parámetros más efectivos para medir la supervivencia de una red y se establece la diversidad como una buena medida de supervivencia. Es por esto que se escoge como base del modelo la diversidad de caminos en lugar de cualquier otro parámetro de la red.

Para calcular esta diversidad se basan en el número de nodos y enlaces que presenta cada camino. Definiendo un camino simple como la combinación de nodos N y enlaces L que presenta dicho camino,

$$P = L \cup N \tag{20}$$

y la longitud de dicho camino $|P|$ como el número total de elementos en L y N. Se puede definir la diversidad entre dos caminos arbitrarios P_a y P_b como:

$$D(P_a, P_b) = 1 - \frac{|P_b \cap P_a|}{|P_a|} \tag{21}$$

Donde la longitud de P_a siempre es igual o menor que la longitud de P_b, y como consecuencia $D(P_a, P_b) = D(P_b, P_a)$.

Nótese que la diversidad de caminos tiene un valor de 1 si los caminos son completamente independientes y un valor de 0 si los caminos son coincidentes. Como ejemplo de la diversidad de caminos se muestran en la Fig. 8 los tres casos posibles entre dos caminos.

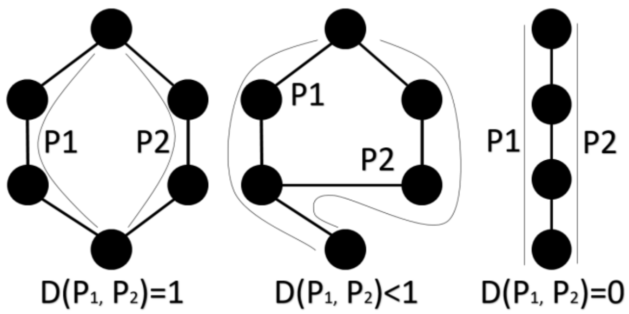


Figura 8 Ejemplo de la diversidad de caminos

En artículos como [2] se usa la diversidad de caminos como herramienta para seleccionar un conjunto de caminos que sean lo más diversos posibles y medir así la efectividad de la diversidad entre cada par de nodos. Sin embargo en este trabajo se propone el uso de la diversidad de caminos para calcular el número de caminos independientes entre cada par de nodos. Cuantos más caminos independientes lleguen a un nodo, más seguro será ante fallos en la red.

Como se comentaba en el punto anterior la diversidad de caminos es una medida muy unida a la supervivencia de las redes [4], es decir, una red con gran diversidad entre sus nodos presenta una supervivencia mayor que otra red con menor diversidad entre sus nodos. Es por esto que se ha cogido este parámetro de la red en lugar de cualquier otro. En el siguiente punto se verá como se relaciona la diversidad de caminos y el IIM para dar lugar al modelo.

2.4 Modelo

Una vez que se han presentado los pilares fundamentales en los que se basa el modelo se van a desarrollar las etapas del mismo. El modelo se compone de cuatro etapas. En primer lugar se obtiene el número de caminos independientes entre cada par de nodos, construyendo una matriz denominada PD. A partir de esta matriz comenzamos a eliminar nodos para ver el impacto que tienen sobre ésta y por tanto sobre la red, calculando así la matriz de impacto IM. Una vez que se tiene la matriz de impacto se tiene que corregir la redundancia que presenta mediante una normalización, y por último se va a introducir la matriz de impacto resultante en el modelo para calcular así la inoperabilidad y en definitiva el riesgo de caída de los nodos. En la Fig. 9 se muestra el flujograma del Modelo.

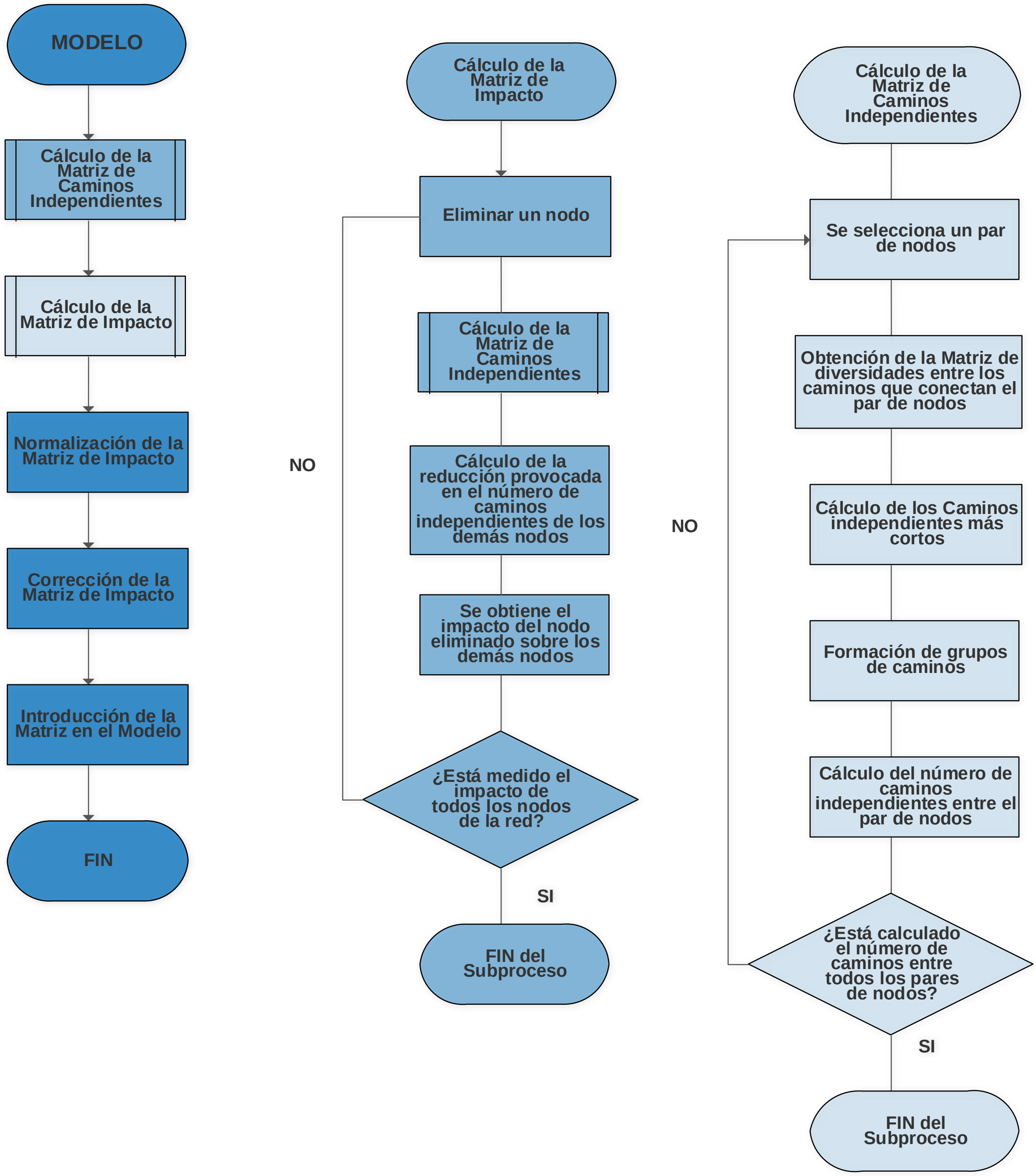


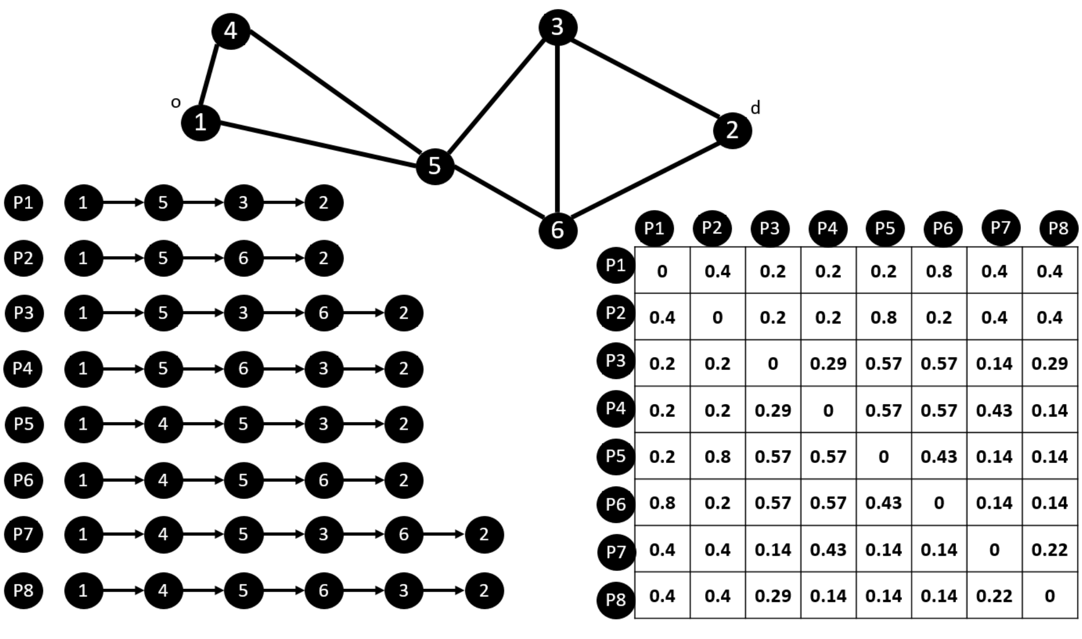
Figura 9 Flujograma del Modelo Propuesto

2.4.1 Matriz de caminos independientes

En esta primera etapa del modelo el objetivo es obtener una matriz cuyos coeficientes representen el número de caminos independientes que existen entre cada par de nodos, haciendo una primera idea de la diversidad que existe entre los nodos de la red.

La programación del modelo se ha llevado a cabo en el programa *Wolfram Mathematica*. Se escogió este programa puesto que posee multitud de funciones aplicables a redes y sobre todo porque tiene una función que devuelve todos los caminos existentes entre dos nodos, y esta función es el punto de partida perfecto para el modelo que se quiere que desarrollar. Para calcular los caminos independientes entre los nodos partiendo del número total de caminos entre cada par de nodos se siguen los siguientes pasos:

1. Crear una matriz $m \, x \, m$, donde m es el número de caminos total entre cada par de nodos, en la que cada coeficiente representa la diversidad entre dos caminos. La diversidad entre dos caminos se calcula mediante la ecuación (21) como veíamos en el punto anterior.
   Lógicamente esta matriz es simétrica puesto que trabajamos con grafos no dirigidos y los caminos entre un nodo y otro coinciden en ambas direcciones. Como consecuencia $D(P_a, P_b) = D(P_b, P_a)$.
   En la Fig. 10 se muestra un ejemplo de la matriz de diversidad. En primer lugar se muestran todos los caminos posibles entre el nodo de origen "o" y el nodo de destino "d" y a la derecha se muestra la matriz de diversidades.



	P1	P2	P3	P4	P5	P6	P7	P8
P1	0	0.4	0.2	0.2	0.2	0.8	0.4	0.4
P2	0.4	0	0.2	0.2	0.8	0.2	0.4	0.4
P3	0.2	0.2	0	0.29	0.57	0.57	0.14	0.29
P4	0.2	0.2	0.29	0	0.57	0.57	0.43	0.14
P5	0.2	0.8	0.57	0.57	0	0.43	0.14	0.14
P6	0.8	0.2	0.57	0.57	0.43	0	0.14	0.14
P7	0.4	0.4	0.14	0.43	0.14	0.14	0	0.22
P8	0.4	0.4	0.29	0.14	0.14	0.14	0.22	0

Figura 10 Matriz de diversidad entre caminos

Por ejemplo, el camino P_3 y el camino P_6 tienen 2 nodos en común y 1 enlace, teniendo 7 elementos el camino de menor longitud. Por tanto la diversidad entre ambos es:

$$D(P_3, P_6) = 1 - \frac{|P_3 \cap P_6|}{|P_3|} = 1 - \frac{3}{7} = 0.57$$

Procediendo igual con el resto de caminos es como obtenemos la matriz de diversidades entre el par de nodos.

2. Una vez que hemos calculado la diversidad entre todos los caminos del par de nodos, el segundo paso es calcular los caminos más cortos independientes. Este paso se realiza partiendo de una función del *Wolfram Mathematica* que proporciona los caminos independientes entre un par de nodos.
   Partiendo del primer camino independiente que da el programa se tiene que recorrer la matriz de diversidades buscando caminos más cortos que sean dependientes a este e independientes al resto de caminos independientes que daba el programa. Si existe algún valor que cumpla estos requisitos se sustituye el primer camino independiente por el otro camino más corto.
   Haciendo esto con el resto de caminos que da el programa conseguimos el conjunto de caminos más cortos e independientes entre sí. La razón de escoger los caminos independientes más cortos tiene que ver con el siguiente paso, en el que se forman grupos de caminos en función de éstos.

   A modo de ejemplo del paso 2 se muestra la Fig. 11. El programa nos proporciona los caminos independientes IP1, IP2 e IP3, sin embargo existen caminos más cortos e independientes como es el caso de P1, que es dependiente de IP1, pero independiente de IP2 e IP3 al igual que IP1. Por tanto escogeremos P1 en lugar de IP1 para el conjunto de caminos independientes de cara al siguiente paso.

3. Una vez que se tiene el conjunto de caminos independientes se forman en este paso grupos de caminos. Se van repasando todos los caminos y comparándolos con el conjunto de caminos independientes. Agruparemos cada camino con el camino independiente con el que más elementos tenga en común, es decir, con el camino independiente con el que su diversidad sea menor. En el ejemplo de la Fig. 11 se agruparía P1 con IP1 y por otro lado IP2 e IP3 formarían otros dos grupos. En el ejemplo mostrado en la Fig. 10 sin embargo, sólo existe un grupo pues no hay dos caminos independientes entre “o” y “d”.

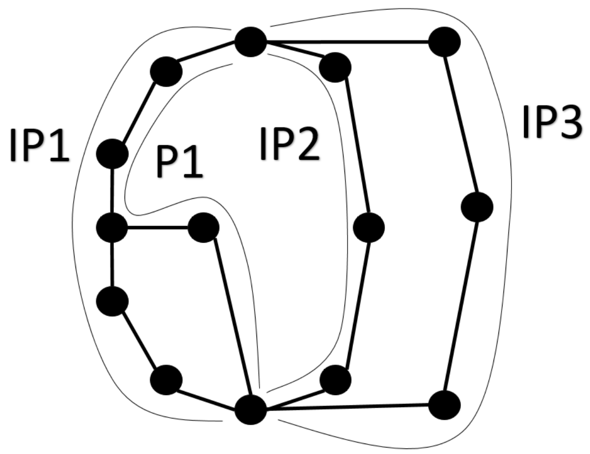


Figura 11 Ejemplo de encontrar los caminos independientes más cortos

4. Con el conjunto de caminos agrupados ya se está en disposición de calcular el número de caminos independientes entre un nodo i y un nodo j como:

$$PD_{ij} = g + \sum_{k=1}^{g} \frac{D_{max}^{k}}{g} \tag{22}$$

   Donde g es el número de grupos y D_{max}^{k} el máximo de la diversidad entre dos caminos que pertenecen al grupo k.

5. Realizamos estos pasos para cada par de nodos y creamos una matriz $n\ x\ n$, llamada matriz de caminos independientes PD. Los coeficientes de esta matriz representan el número de caminos independientes entre cada par de nodos. Lógicamente cuanto mayor sea este número mejor diversidad habrá entre los nodos. Además también se trata de una matriz simétrica puesto que los caminos independientes de A a B son los mismos que de B a A al trabajar con grafos no dirigidos como se ha comentado anteriormente.

Después de calcular la matriz de caminos independientes PD se pueden sacar varias conclusiones. En primer lugar podemos calcular el número de caminos independientes que llega a cada nodo desde los demás, que se calcula como la suma de los coeficientes a lo largo de las filas o las columnas de PD. Esto es:

$$\sum_{i=1}^{n} PD_{ij} = PD_j = PD_i = \sum_{j=1}^{n} PD_{ij} \tag{23}$$

Se puede usar este número como medida de la supervivencia de un nodo. Cuantos más caminos independientes lleguen a un nodo, más probable será que éste permanezca conectado a la red tras fallos en la misma. Por tanto con estos valores podemos hacer un primer ranking de los nodos más fuertes y más vulnerables de la red.

Otra información útil que aporta esta matriz es el nivel de diversidad que existe entre los nodos. Cuanto mayor sea el coeficiente mejor conectados estarán los nodos y como consecuencia la probabilidad de que ambos nodos permanezcan conectados tras un fallo también será mayor. En cambio, un bajo valor del coeficiente indica que en caso de un fallo en la red, es muy probable que se rompa la conexión entre ambos nodos.

Se ha llegado a una matriz que aporta información de la diversidad entre los nodos, pero no aporta información alguna sobre el impacto que causan unos nodos sobre otros. Recordemos que el objetivo es llegar a una matriz de impacto que se pueda introducir en el IIM en lugar de la matriz S que se calculaba mediante juicio de expertos.

2.4.2 Matriz de Impacto

En esta segunda etapa del modelo se muestra como calcular la matriz de impacto a partir de la matriz de caminos independientes.

Para el cálculo del impacto que un nodo provoca sobre los demás se va a proceder a eliminar dicho nodo y se va a ver como afecta a la red. Este paso se tiene que hacer tantas veces como nodos haya en la red para calcular el impacto de todos. El impacto de un nodo sobre los demás se va a medir como la reducción que provoca en los caminos independientes de los demás nodos la caída de dicho nodo.

Para medir este impacto se comienza eliminando el nodo 1 y se realiza todo el procedimiento explicado en la etapa anterior hasta llegar a una segunda matriz de caminos independientes en el que el nodo 1 está eliminado. Denotaremos la matriz de caminos independientes al eliminar el nodo n como, PD^n. Y de la misma manera que antes, sumamos los coeficientes de las filas o las columnas llegando al número de caminos independientes que llegan a cada nodo, número que se habrá visto reducido ante la caída de un nodo.

$$\sum_{i=1}^{n} PD^{n}{}_{ij} = PD^{n}{}_{j} \tag{24}$$

Con este valor se calcula el impacto del nodo eliminado n sobre el nodo j como:

$$IM_{jn} = \frac{PD_j - PD^{n}{}_{j}}{PD_j} \tag{25}$$

Donde IM_{jn} representa el porcentaje de caminos independientes perdido por el nodo j cuando el nodo n está totalmente inoperable. Por ejemplo, un coeficiente de $IM_{jn} = 0.8$ indica que si el nodo n cae, el número de caminos independientes que llega a j se verá reducido en un 80%. Viendo el impacto sobre el resto de nodos se logra obtener la columna n-ésima de la matriz de impacto. Siendo lógicamente 1 los coeficientes de la diagonal puesto que representan el impacto sobre sí mismo.

Esto ya se va pareciendo más a los coeficientes de la matriz de influencia del IIM, que se correspondían con la inoperabilidad que n provoca en j cuando el primero está completamente inoperable.

Eliminando cada nodo de la red y calculando el impacto que tiene sobre el resto de nodos, llegamos a la matriz de impacto IM, donde cada coeficiente respresenta el impacto relativo que un nodo tiene sobre otro, es decir, cada porcentaje está calculado sobre el número de caminos que le llegan a un nodo. Debido a esto no se pueden realizar comparaciones entre las distintas filas de la matriz. Imaginémos que existe un nodo A al que le llegan 20 caminos independientes, y un nodo B al que le llegan 2 caminos independientes. Si el impacto que provoca un tercer nodo C sobre A es de 0.7 y el impacto sobre B es de 0.5 y el nodo C queda inoperable, se pensará en proteger A en lugar de B, sin embargo, el nodo B será más crítico pues el número de caminos independientes que le llega será 1 y por tanto será vulnerable ante el próximo fallo de la red. En cambio el nodo A aunque haya sufrido un impacto mayor le quedarán aún 6 caminos independientes que lleguen a él y por tanto no será crítico de momento.

Esto nos lleva a pensar, que aunque la matriz de impacto nos de una información valiosa por filas y nos ayude a calular la reducción total de caminos independientes en cada nodo, se necesitará de un valor absoluto para poder comparar los diferentes nodos de la red como se verá más adelante.

Otra información útil que si podemos sacar de la matriz de impacto son el índice de dependencia, por el que veremos los nodos más débiles de la red y la influencia ganada, por el que veremos los nodos que más impactan sobre la red.

2.4.3 Normalización de la matriz de impacto

Nótese que la matriz de impacto presenta una alta redundancia en sus filas, debido a que los caminos independientes están formados por multitud de nodos, y cada vez que eliminamos un nodo de ese camino, se está eliminando el mismo camino. Por tanto si suponemos que caen dos nodos que formaban el mismo camino, se está duplicando la reducción de caminos independientes y en consecuencia la matriz de impacto. En el caso de que se trabajara simplemente con fallos aislados de nodos, la matriz se podría incluir en el modelo y no estaríamos incurriendo en redundancia alguna. No obstante, puesto que se va a trabajar con la degradación de todos los componentes al mismo tiempo se tiene que decrementar la redundancia presentada en dicha matriz.

El método que se ha usado para eliminar la redundancia es la normalización de las filas de la matriz de impacto, asumiendo por tanto que solo se puede impactar en un 100% en cada nodo. Los coeficientes de la nueva matriz de impacto se calcularán para los elementos en los que $i \neq j$, es decir, para los elementos que no pertenecen a la diagonal, como:

$$IM'_{ij} = \frac{IM_{ij}}{\sum_{j=1}^{j=i} IM_{ij}} \quad (26)$$

Mientras que para los elementos de la diagonal se seguirá manteniendo la unidad, puesto que el efecto de la degradación de un nodo sobre sí mismo es siempre igual.

Tras eliminar la redundancia se observa que para el caso que todos los nodos se degradan al mismo tiempo el modelo funciona de manera correcta, pero para los casos en los que un nodo presenta mayor degradación que el resto el comportamiento de la red no es el esperado pues habría que utilizar la matriz de impacto sin normalizar.
Aquí se presenta un dífícl problema de resolver, y es que al normalizar la matriz se pierde gran parte de la información y se falsean los datos de la matriz de impacto. Por tanto, si se quiere tener en cuenta ambos efectos hay que llegar a un equilibrio de ambas matrices. La solución propuesta es una combinación de las dos matrices de la siguiente manera.

En función de la degradación que tenga un componente con respecto a la media de todos los componentes de la red, se tomarán los datos de una u otra matriz, es decir, en el caso de que un componente presente una degradación igual o menor que la media de la degradación de todos los componentes de la red, se tomará el impacto correspondiente a la matriz de impacto normalizada:

$$Si \;\; F_i(t) \leq \overline{F(t)}, \qquad IM''_{ij} = IM'_{ij} \quad (27)$$

En segundo lugar, si la degradación de un componente es mayor que la media, se tomará un impacto combinado entre la matriz de impacto sin normalizar y la matriz de impacto normalizada de la siguiente manera:

$$Si \;\; F_i(t) > \overline{F(t)}, \qquad IM''_{ij} = \left(F_i(t) - \overline{F(t)}\right) \cdot \; IM_{ij} \cdot FC + \overline{F(t)} \cdot IM'_{ij} \quad (28)$$

Donde FC es un factor corrector que corresponde con la relación entre el número de componentes que están por debajo de la media y el número total de componentes. De esta manera, cuantos más componentes haya por encima de la media, más cerca se estará de la matriz de impacto normalizada, y cuantos menos haya, se estará en un caso más parecido al que se degrada únicamente un nodo.

De esta manera conseguimos una matriz de impacto dinámica en función de las condiciones que se presenten en la red sin requerir tiempo ni esfuerzo prácticamente. Esta matriz ya está lista para incorporarse al IIM, lo cual se hará en la última etapa.

2.4.4 Incorporación al IIM

Volviendo al modelo de la ecuación (18), ya se disponen de las herramientas necesarias para calcular la inoperabilidad de los nodos y por tanto poder clasificarlos por su supervivencia.
Como se ha mencionado anteriormente el IIM sólo se utiliza para fallos catastróficos sin tener en cuenta la degradación de los nodos, sin embargo, se quiere que el modelo propuesto tenga en cuenta ambos. Para darle solución se divide el término c en dos sumandos:

$$c(t) = F(t) + e(t) \tag{29}$$

Donde $F(t)$ representa la probabilidad de fallo de cada nodo, y $e(t)$ representa la inoperabilidad de cada nodo debida a los fallos y las perturbaciones externas a la red. Tras esta modificación, la ecuación (18) se puede expresar como:

$$x(t) = IM \cdot (F(t) + e(t)) \tag{30}$$

Teniendo en cuenta que el modelo proporciona la reducción en porcentaje del número de caminos que llegan a un nodo, para encontrar aquellos con mayor riesgo de quedar separados de la red, hay que calcular el número de caminos independientes que no han caído. Si se multiplican los caminos que llegan a un nodo al inicio PD_i por la inoperabilidad de dicho nodo $x(t)$, nos quedan los caminos que han caído. Por tanto para evaluar el riesgo en cada nodo hay que multiplicar el número de caminos totales que llega a un nodo por el porcentaje que caminos que quedan disponibles, que es $1 - x\ (t)$. De esta manera el número de caminos que quedan disponibles (PR) será:

$$PR_i(t) = PD_i \cdot (1 - x_i) \tag{31}$$

Después de calcular los caminos disponibles que llegan a cada nodo, el que presente mayor riesgo, y por tanto el más crítico, será aquel con un menor valor de caminos disponibles. Este nodo será al que se preste más atención.
Nótese que la probabilidad de fallo o degradación de los nodos dependen del tiempo, al igual que que los fallos externos. Sin embargo, la matriz de impacto y la matriz de impacto normalizada, son puramente topológicas y no dependen del tiempo, ya que las conexiones entre los nodos permanecerán constantes. Si la configuración de la red cambia hay que volver a calcular la matriz

de impacto. Además se ha investigado la posibilidad de no sólo basar la matriz de impacto en la topología, sino también en otros parámetros como la capacidad y los resultados son igualmente prometedores.

3 CASO PRÁCTICO

Este capítulo muestra un caso práctico al que se le ha aplicado el modelo propuesto en el trabajo. Aunque el modelo funciona para todo tipo y tamaño de redes, se ha supuesto una red de 17 equipos, todos comunicados y que interaccionan todos con todos, en la que se quiere ver en diferentes casos y a lo largo del tiempo cuales son los puntos más débiles de la red de cara a mejorar la eficiencia del mantenimiento en los mismos. Partiendo de una base común, que es la matriz de impacto, se pueden calcular todos los casos que se requieran en un tiempo pequeño, lo cual hace el modelo muy práctico para tener controladas las redes en todo momento. Tras aplicar el modelo se obtienen los resultados de los distintos casos y se analizan para sacar conclusiones.

3.1 Introducción

Una vez desarrollado el modelo, se procede a aplicar el mismo a la red de la Fig. 12. A la hora de aplicar el modelo a un caso práctico se diferencian dos partes. Por un lado la obtención de la matriz de impacto, la cual es común a todos los casos de estudio de la red propuesta, y por otro lado la situación dependiente del tiempo que se quiera plantear. La primera parte se lleva a cabo mediante un programa en *Wolfram Mathematica*, mientras que la segunda parte se hace de manera interactiva con programas como Excel o Xcelsius, pudiéndose implementar incluso en SCADA para tener una visión contínua a lo largo del tiempo.

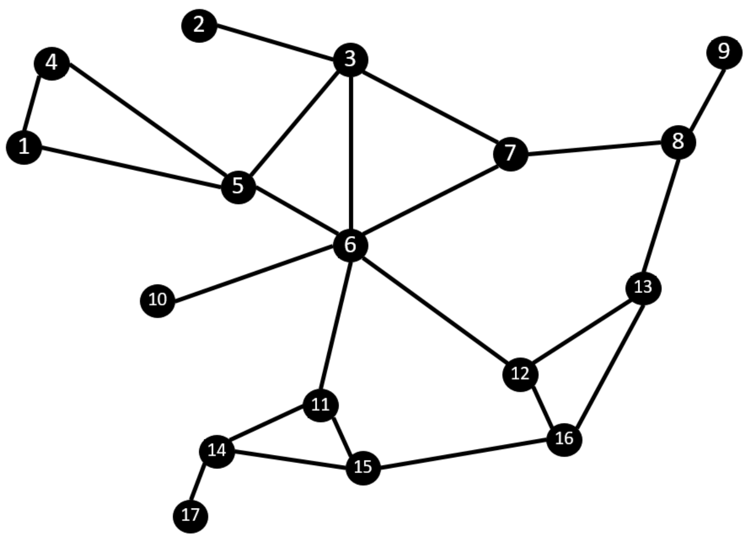


Figura 12 Red de estudio

Se trata de una red con 17 equipos distribuidos de manera diferente. Existen equipos más centralizados como el 6, otros intermedios como el 5, el 11 o el 7 y otros en la periferia como el 9 o el 17. Se supone que en el instante inicial la degradación de los componentes es nula puesto que todos son nuevos. En los puntos sucesivos se procede a aplicar las distintas etapas del modelo.

3.2 Aplicación del modelo

El primer paso del modelo es calcular la matriz de caminos independientes (PD) de la red. Como ejemplo se calculará el número de caminos independientes entre el nodo 6 y el 16, y para el resto de par de nodos se hará de la misma manera.

En la Fig. 13 se presentan todos los caminos posibles entre el par de nodos ordenados de menor a mayor longitud. Este es el punto de partida de nuestro modelo, que proviene como se ha comentado del programa *Wolfram Mathematica*.

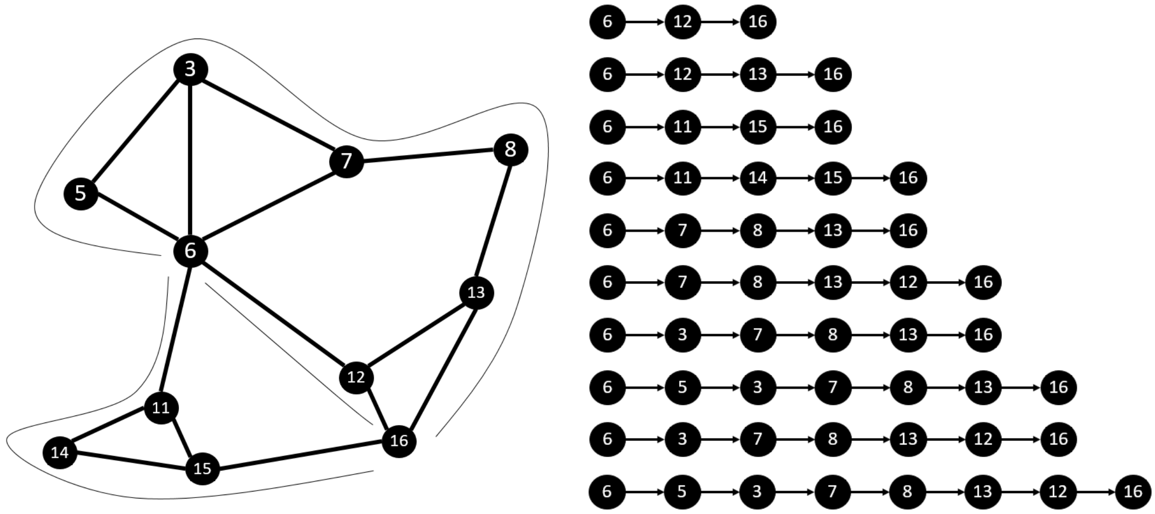


Figura 13 Caminos entre el nodo 6 y el nodo 16

Una vez que se tienen todos los caminos, el segundo paso es calcular la diversidad entre ellos construyendo una matriz $p\,x\,p$, siendo p el número de caminos entre el par de nodos. Como se explicaba en el capítulo anterior se ha utilizado la ecuación de la diversidad propuesta en [5], y como ejemplo se muestra el cálculo entre los caminos $P_2(6{,}12{,}13{,}16)$ y $P_5(6{,}7{,}8{,}13{,}16)$ de la Fig. 13.

$$D(P_2,P_5) = 1 - \frac{|P_2 \cap P_5|}{|P_2|} = 1 - \frac{2}{5} = 0.6$$

Haciendo este cálculo con todos los pares de caminos posibles se crea la matriz de diversidades que se muestra en la Tabla I, que como se puede observar, es simétrica. Cuanto mayor sean los valores de la matriz, habrá más diversidad entre el par de nodos y por tanto la conexión entre ellos será más segura. Los coeficientes de la matriz con un 1 representan que ambos caminos son independientes.

En principio se pueden distinguir tres agrupaciones de caminos, que como se verá más adelante constituirán los diferentes grupos de cara al cálculo de los caminos independientes.

Tabla I

DIVERSIDAD ENTRE CAMINOS

PATHS	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	-	0.33	1.00	1.00	1.00	0.33	1.00	1.00	0.33	0.33
P2	0.33	-	1.00	1.00	0.60	0.40	0.60	0.60	0.40	0.60
P3	1.00	1.00	-	0.20	1.00	1.00	1.00	1.00	1.00	1.00
P4	1.00	1.00	0.20	-	1.00	1.00	1.00	1.00	1.00	1.00
P5	1.00	0.60	1.00	1.00	-	0.14	0.14	0.14	0.29	0.29
P6	0.33	0.40	1.00	1.00	0.14	-	0.44	0.44	0.11	0.11
P7	1.00	0.60	1.00	1.00	0.14	0.44	-	0.11	0.11	0.22
P8	1.00	0.60	1.00	1.00	0.14	0.44	0.11	-	0.36	0.09
P9	0.33	0.40	1.00	1.00	0.29	0.11	0.11	0.36	-	0.09
P10	0.33	0.40	1.00	1.00	0.29	0.11	0.22	0.09	0.09	-

Tras la obtención de dicha matriz el siguiente paso es calcular los caminos independientes más cortos. A partir de los caminos independientes proporcionados por el programa *Wolfram Mathmatica*, que se muestran en la Fig. 14, habrá que comprobar si existen caminos independientes más cortos de cara al siguiente paso del modelo.

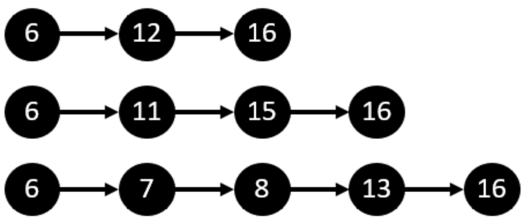


Figura 14 Caminos independientes proporcionados por el programa

Partiendo de los 3 caminos independientes de la Fig. 14, que se corresponden con P_1, P_4 y P_5, se va a ir comprobando uno por uno si existen caminos más cortos por los que se puedan sustituir. El primer camino independiente es el más corto entre el nodo 6 y el nodo 16, por lo que no existe la posibilidad de que se pueda sustituir por uno más corto. Si nos fijamos en el segundo camino independiente y en la matriz de diversidades, se aprecia que existe un camino P_3 que es más corto que P_4 y que es independiente además del resto de caminos independientes, P_1 y P_5, por lo tanto cumple todos los requisitos. Por úlitmo, si nos fijamos en el tercer camino independiente, no existe ningún otro camino que cumpla las especificaciones que deseamos. Finalmente el conjunto de caminos más cortos independientes es P_1, P_3 y P_5.

Tras obtener el conjunto de caminos anterior se está en disposición de realizar el siguiente paso del modelo, la agrupación de los caminos. Recordemos que este paso consistía en agrupar cada camino con el camino más corto independiente con el que tuviera más elementos en común y en consecuencia, menor diversidad. Por ejemplo, aunque el camino P_2 se podría agrupar con P_1 y con P_5, se agrupa con el camino P_1 pues la diversidad con este es 0.33 frente a la diversidad de 0.6 que tiene con el camino P_5. En la Fig. 15 se muestran los distintos grupos que han surgido tras la clasificación de los caminos.

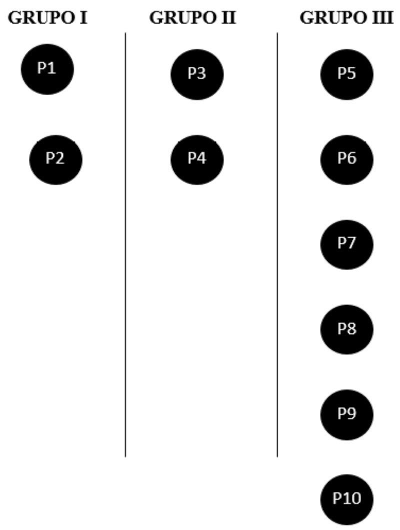


Figura 15 Grupos de Caminos

Puesto que no existe ningún algoritmo para calcular el número exacto de caminos independientes entre un par de nodos se trataron de utilizar métodos directos como la media de las diversidades o la suma de todas las diversidades, pero se observó que no aproximaban bien el número total de caminos independientes, generlamente siempre ofrecía un número menor. Se propuso un nuevo método en el que se contaban por un lado los caminos indpendientes, que en este caso equivalen al número de grupos, y por otro la ponderación con respecto al número de grupos, de los máximos de diversidad entre dos caminos de un mismo grupo. Aunque realmente el número de caminos independientes es algo mayor, se trata de una buena aproximación.

$$PD_{6-16} = g + \sum_{k=1}^{g} \frac{D_{max}^{k}}{g}$$

Aplicando la ecuación al caso de estudio, el número de grupos se puede sacar de manera directa puesto que es igual al número de caminos independientes. En este caso $g = 3$. Sin embargo, el cálculo de D_{max}^{k} no es tan sencillo por lo general. En el caso de los dos primeros grupos, al haber solo 2 caminos el cálculo de D^1_{max} y D^2_{max} sí que es directo e igual a la diversidad entre dichos caminos. Es decir, $D^1_{max} = D(P_1, P_2) = 0.33$ y $D^2_{max} = D(P_3, P_4) = 0.2$. En el caso de D^3_{max} no es tan fácil puesto que hay 6 caminos. El procedimiento es extraer la parte de la matriz que haga referencia a estos caminos y encontrar el máximo. Dicha matriz se muestra en la Tabla II. Como se puede apreciar, el máximo de las diversidades entre los caminos del tercer grupo corresponde con $D^3_{max} = 0.44$. Teniendo ya todos los datos se puede calcular el número de caminos independientes entre el nodo 6 y el 16.

$$PD_{6-16} = 3 + \frac{0.33}{3} + \frac{0.2}{3} + \frac{0.44}{3} = 3.326$$

Tabla II

DIVERSIDADES ENTRE LOS CAMINOS DEL GRUPO III

PATHS	P5	P6	P7	P8	P9	P10
P5	-	0.14	0.14	0.14	0.29	0.29
P6	0.14	-	0.44	0.44	0.11	0.11
P7	0.14	0.44	-	0.11	0.11	0.22
P8	0.14	0.44	0.11	-	0.36	0.09
P9	0.29	0.11	0.11	0.36	-	0.09
P10	0.29	0.11	0.22	0.09	0.09	-

De esta manera hemos calculado uno de los coeficientes de la matriz de caminos independientes PD. Haciendo lo mismo para el resto de pares de nodos de la red se obtiene la matriz completa, la cual se muestra en la Tabla III. Como ya se ha comentado en varias ocasiones, la matriz de caminos independientes es una matriz simétrica al trabajar con redes no dirigidas.

Esta matriz al ser el caso inicial, es puramente topológica. Es decir, no existe degradación alguna ni fallos externos que influyan sobre los caminos independientes entre los nodos. A partir de aquí, según los acontecimientos irán disminuyendo, pero de esta matriz ya se pueden sacar las primeras conclusiones.

Con respecto a los coeficientes de la matriz, son en cierto modo una medida de la conectividad entre los pares de nodos. Cuanto mayor sea el coeficiente mejor conectados estarán los nodos y más probabilidad tendrán de asegurar su conexión ante fallos en la red. Por ejemplo, el nodo 13 y el 16 están conectados con una diversidad de caminos de 3.53, lo que implica que pueden caer varios nodos y la conexión se seguirá asegurando entre ambos. En cambio, un coeficiente pequeño implica que ante fallos en la red hay una alta probabilidad de que se desconecten los nodos. Por ejemplo, entre los nodos 6 y 2 hay 1.60 caminos independientes, lo que implica que ante la caída de un nodo específico no se asegura que sigan conectados dichos nodos.

Tabla III

MATRIZ DE CAMINOS INDEPENDIENTES

NODES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0.00	1.57	1.80	2.00	2.00	1.80	1.86	1.89	1.73	1.57	1.86	1.91	1.91	1.89	1.91	1.91	1.73
2	1.57	0.00	1.00	1.57	1.33	1.60	1.57	1.67	1.55	1.43	1.71	1.78	1.73	1.78	1.82	1.82	1.64
3	1.80	1.00	0.00	1.80	2.17	3.11	2.40	2.69	1.67	1.60	2.73	2.72	2.82	2.65	2.82	2.86	1.78
4	2.00	1.57	1.80	0.00	2.00	1.80	1.86	1.89	1.73	1.57	1.86	1.91	1.91	1.89	1.91	1.91	1.73
5	2.00	1.33	2.17	2.00	0.00	2.17	2.47	2.61	1.71	1.33	2.70	2.76	2.87	2.64	2.76	2.89	1.73
6	1.80	1.60	3.11	1.80	2.17	0.00	3.25	2.47	1.78	1.00	2.33	3.30	3.33	2.63	2.47	3.33	1.60
7	1.86	1.57	2.40	1.86	2.47	3.25	0.00	2.36	1.55	1.71	2.79	2.79	2.44	2.73	2.79	2.54	1.82
8	1.89	1.67	2.69	1.89	2.61	2.47	2.36	0.00	1.00	1.78	2.63	2.63	2.36	2.67	2.73	2.53	1.82
9	1.73	1.55	1.67	1.73	1.71	1.78	1.55	1.00	0.00	1.64	1.82	1.71	1.55	1.82	1.78	1.71	1.64
10	1.57	1.43	1.60	1.57	1.33	1.00	1.71	1.78	1.64	0.00	1.55	1.82	1.82	1.60	1.71	1.78	1.43
11	1.86	1.71	2.73	1.86	2.70	2.33	2.79	2.63	1.82	1.55	0.00	2.62	2.64	2.17	3.14	2.47	1.33
12	1.91	1.78	2.72	1.91	2.76	3.30	2.79	2.63	1.71	1.82	2.62	0.00	3.29	2.57	2.59	3.26	1.71
13	1.91	1.73	2.82	1.91	2.87	3.33	2.44	2.36	1.55	1.82	2.64	3.29	0.00	2.62	2.53	3.53	1.78
14	1.89	1.78	2.65	1.89	2.64	2.63	2.73	2.67	1.82	1.60	2.17	2.57	2.62	0.00	2.17	2.63	1.00
15	1.91	1.82	2.82	1.91	2.76	2.47	2.79	2.73	1.78	1.71	3.14	2.59	2.53	2.17	0.00	2.33	1.33
16	1.91	1.82	2.86	1.91	2.89	3.33	2.54	2.53	1.71	1.78	2.47	3.26	3.53	2.63	2.33	0.00	1.60
17	1.73	1.64	1.78	1.73	1.73	1.60	1.82	1.82	1.64	1.43	1.33	1.71	1.78	1.00	1.33	1.60	0.00
PD_i	**29.33**	**25.56**	**36.61**	**29.33**	**36.15**	**37.96**	**36.93**	**35.73**	**26.37**	**25.34**	**36.35**	**39.37**	**39.13**	**35.45**	**36.79**	**39.11**	**25.66**

Si sumamos los coeficientes a lo largo de las filas o las columnas obtenemos el número de caminos independientes que llegan a un nodo desde el resto PD_i, los cuales se representan al final de la Tabla III. A partir de estos valores se puede realizar una primera clasificación según la diversidad de los caminos que llegan a los distintos nodos, viendo así cuales son los nodos más vulnerables y cuales son aquellos que tendrán más probabilidad de permanecer unidos a la red a pesar de los fallos.

En el caso de estudio se podría pensar que al ser el nodo 6 el más central sería el que mejor estuviera contectado con el resto de nodos, sin embargo no es así, los nodos a los que más caminos independientes llegan son el 12, el 13 y el 16, con una suma cercana a 40.

Otra cuestión a tener en cuenta es que el número de caminos que llegan a un nodo no implica la diversidad de los mismos. Por ejemplo, aunque al nodo 4 le llegan 600 caminos diferentes desde el resto de nodos, el número de caminos independientes que le llegan es de 29.33. Sin embargo al nodo 16 le llegan 39.11 caminos independientes recibiendo la mitad de caminos totales, 300.

El siguiente paso, una vez que se ha obtenido la matriz de caminos independientes, es calcular la matriz de impacto para introducirla en el modelo y calcular los distintos casos de estudio. Para ello, como se indicó en el desarrollo del modelo se va a proceder a eliminar los distintos nodos por separado y a cuantificar el impacto que tienen sobre los demás nodos de la red.

A continuación se muestra a modo de ejemplo el cálculo de una de las columnas de la matriz de impacto, concretamente el caso en el cual se ha eliminado el nodo 12. Para el resto de nodos se haría de la misma manera. En la Fig. 16 se muestra como queda la nueva red tras eliminar el nodo 12.

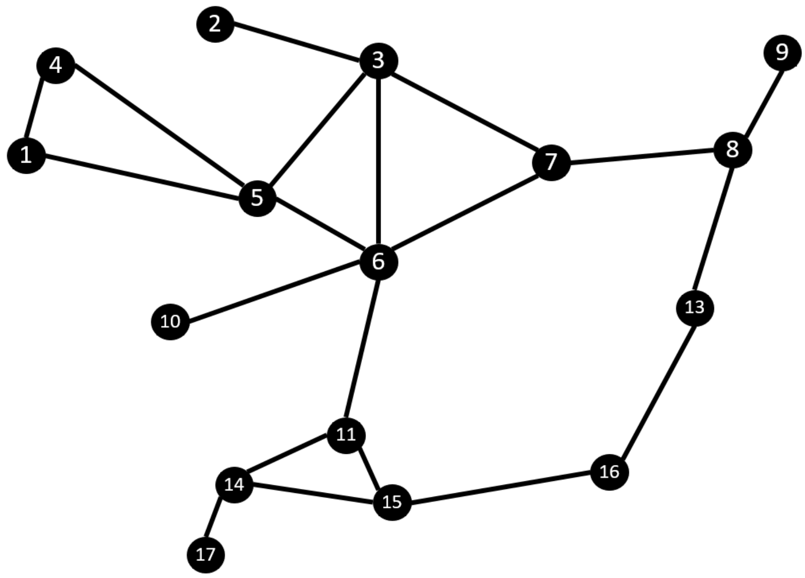


Figura 16 Red de estudio tras eliminar el punto 12

El proceso que se sigue para calcular la matriz de caminos independientes en el nuevo caso es el mismo que el del caso inicial, por lo que se muestra directamente la suma de los caminos independientes de cada nodo en la Tabla IV.

Comparando los resultados con los del caso inicial se puede apreciar una reducción en el número de caminos independientes que llegan a cada nodo, estén conectados directa o indirectamente con el nodo 12. Esto tiene sentido, pues como mínimo han perdido los caminos independientes que partían del nodo 12 a los demás. Además habrán perdido los caminos que pasaran de manera intermedia por el nodo eliminado.

Con la suma de caminos independientes calculada, estamos en disposición de calcular en este caso la columna 12 de la matriz de impacto como la reducción porcentual en dicha suma.

Se muestra a continuación el cálculo del coeficiente IM_{5-12}, impacto que tiene el nodo 12 sobre el nodo 5. Para el resto de coeficientes IM_{x-12} la operación será la misma sólo sustituyendo por los números que corresponden. Al final de la Tabla IV se muestra el impacto que tiene el nodo 12 sobre el resto de nodos.

$$IM_{5-12} = \frac{PD_5 - PD^{12}{}_5}{PD_5} = \frac{36.15 - 31.4}{36.15} = 0.13$$

Tabla IV

SUMA DE CAMINOS INDEPENDIENTES TRAS ELIMINAR EL NODO 12

NODOS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$PD^{12}{}_i$	27.4	23.7	31.3	27.4	31.4	31.1	31.5	30.2	24.0	23.3	30.6	0	30.3	30.5	31.3	30.5	23.8
IMPACTO	7%	7%	15%	7%	13%	18%	15%	16%	9%	8%	16%	100%	23%	14%	15%	22%	7%

Lo mismo que se ha hecho con el nodo 12 se tiene que hacer con el resto de nodos para obtener la matriz de impacto completa. Por tanto, se tiene que aplicar el modelo tantas veces como número de nodos se tenga, además del caso inicial. La matriz de impacto se muestra en la Tabla V, en la que cada columna representa el impacto de un nodo sobre el resto. Lógicamente los coeficientes de la diagonal son 1 puesto que representa la degradación del propio nodo.

Tabla V

MATRIZ DE IMPACTO

NODES	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	δ_i
1	100%	5%	21%	21%	97%	38%	10%	14%	6%	5%	9%	7%	11%	13%	11%	11%	6%	284%
2	8%	100%	100%	8%	22%	41%	12%	17%	6%	6%	10%	7%	14%	14%	12%	12%	6%	294%
3	6%	3%	100%	6%	21%	57%	29%	28%	5%	4%	17%	15%	26%	13%	19%	21%	5%	275%
4	21%	5%	21%	100%	97%	38%	10%	14%	6%	5%	9%	7%	11%	13%	11%	11%	6%	284%
5	8%	4%	39%	8%	100%	54%	26%	27%	5%	4%	16%	13%	24%	12%	19%	22%	5%	287%
6	6%	4%	23%	6%	19%	100%	25%	24%	5%	3%	21%	18%	25%	13%	25%	28%	4%	248%
7	6%	4%	23%	6%	19%	57%	100%	30%	4%	5%	18%	15%	30%	13%	18%	20%	5%	273%
8	6%	5%	20%	6%	19%	54%	37%	100%	3%	5%	19%	16%	39%	13%	18%	21%	5%	284%
9	7%	6%	18%	7%	21%	41%	21%	100%	100%	6%	13%	9%	24%	13%	12%	13%	6%	317%
10	8%	6%	20%	8%	21%	100%	15%	19%	6%	100%	12%	8%	16%	14%	16%	17%	6%	290%
11	6%	5%	18%	6%	18%	45%	20%	24%	5%	4%	100%	16%	23%	15%	41%	37%	4%	286%
12	5%	5%	18%	5%	17%	51%	27%	28%	4%	5%	26%	100%	30%	14%	25%	30%	4%	295%
13	5%	4%	18%	5%	18%	54%	28%	28%	4%	5%	26%	23%	100%	13%	23%	29%	5%	285%
14	6%	5%	18%	6%	18%	45%	19%	22%	5%	5%	43%	14%	22%	100%	44%	33%	3%	306%
15	6%	5%	17%	6%	18%	51%	20%	23%	5%	5%	40%	15%	22%	15%	100%	34%	4%	284%
16	5%	5%	17%	5%	17%	53%	25%	27%	4%	5%	27%	22%	28%	13%	24%	100%	4%	281%
17	7%	6%	16%	7%	21%	33%	11%	16%	6%	6%	25%	7%	11%	100%	28%	20%	100%	320%
ρ_j	116%	76%	408%	116%	462%	811%	334%	440%	79%	76%	331%	210%	356%	299%	345%	357%	77%	

También se muestra en esta tabla la ganancia de influencia y el índice de dependencia, que se calculaban como la suma de los coeficientes a lo largo de las columnas y de las filas respectivamente, sin tener en cuenta el impacto que tienen sobre sí mismo.

A la vista de los resultados se puede observar que el nodo que más impacto tiene en la red es el nodo 6, que corresponde con el nodo que tiene la mayor ganancia de influencia. Por el contrario, el nodo más dependiente del resto de nodos, y por tanto más vulnerable, es el nodo 17.

Viendo la matriz se aprecia que los nodos más centralizados son aquellos que más impactan sobre la red, y los nodos exteriores son los más vulnerables. Parece lógico que la centralidad de los nodos esté relacionada con el impacto que tienen sobre la red, pues cuanto más centrado estén, más caminos pasarán por ellos. Pero no es el único factor. Por ejemplo, aunque los nodos 11 y 12 tienen una centralidad mayor que el 8, éste tiene un impacto mayor sobre la red. Esto se debe a que nodos como el 8, el 3 o el 14 tienen otro nodo que depende totalmente de ellos, y por tanto causan un mayor impacto a la red.

En la Tabla VI se muestran los valores de los principales parámetros de una red, así como el impacto total que tienen los nodos sobre la red.

Tabla VI

PRINCIPALES PARÁMETROS DE LA RED

ID	Impacto	Closeness	Betweenness	Impacto	Closeness	Betweenness
	Valor de los parámetros			Clasificación de los nodos de menor a mayor		
1	1.16	0.31	0.00	10	17	17
2	0.76	0.32	0.00	2	9	9
3	4.08	0.46	19.50	17	1	1
4	1.16	0.31	0.00	9	4	4
5	4.62	0.43	28.00	1	2	2
6	8.11	0.57	65.50	4	14	10
7	3.34	0.44	19.00	12	10	15
8	4.40	0.37	17.50	14	8	13
9	0.79	0.28	0.00	11	15	16
10	0.76	0.37	0.00	7	13	12
11	3.31	0.44	27.00	15	16	14
12	2.10	0.44	14.00	13	5	8
13	3.56	0.38	8.50	16	12	7
14	2.99	0.35	15.00	3	11	3
15	3.45	0.38	8.50	8	7	11
16	3.57	0.39	9.50	5	3	5
17	0.77	0.26	0.00	6	6	6

Como se puede observar en la tabla VI el *closeness centrality* y el *betweenness* siguen la misma línea que el impacto de los nodos sobre la red, aunque ninguno de los dos coincide totalmente como se puede ver en la clasificación de los nodos en función de los parámetros.

Volviendo al modelo, en la Tabla V se puede ver claramente como la suma de las columnas de la matriz de impacto es bastante mayor que 1 y por tanto no se puede aplicar la relación $S = (I - A)^{-1}$, la cual es bastante restrictiva. Sin embargo, para nuestro modelo vamos a tomar la matriz S directamente como la matriz de impacto, pues no sólo tiene en cuenta los impactos de primer orden sino también los de orden superior.

Como se comentaba en el desarrollo del modelo, no se puede utilizar esta matriz directamente como matriz S puesto que en el caso en el que todos los nodos se degradan a la vez se incurre en una alta redundancia que no representa la realidad. Para mitigar este efecto se decide normalizar la matriz, como se puede ver en la Tabla VII, aunque de esta manera se conseguía el efecto contrario, conseguíamos un buen modelo cuando todos los nodos se degradan a la vez pero fallaba cuando había nodos con mayor degradación que el resto. Finalmente se llegó a una buena solución combinando ambas matrices para tener en cuenta los dos efectos.

Tabla VII

MATRIZ DE IMPACTO NORMALIZADA

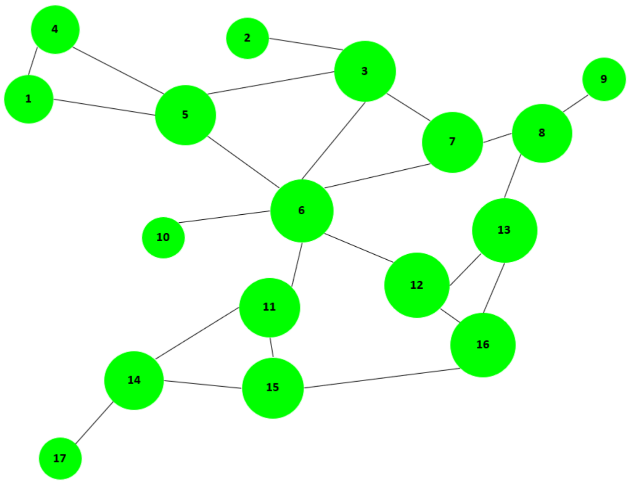
NODES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	100%	2%	8%	8%	34%	13%	3%	5%	2%	2%	3%	2%	4%	4%	4%	4%	2%
2	3%	100%	34%	3%	8%	14%	4%	6%	2%	2%	3%	2%	5%	5%	4%	4%	2%
3	2%	1%	100%	2%	8%	21%	11%	10%	2%	2%	6%	5%	10%	5%	7%	8%	2%
4	8%	2%	8%	100%	34%	13%	3%	5%	2%	2%	3%	2%	4%	4%	4%	4%	2%
5	3%	1%	14%	3%	100%	19%	9%	9%	2%	1%	6%	5%	8%	4%	7%	8%	2%
6	2%	2%	9%	2%	8%	100%	10%	10%	2%	1%	9%	7%	10%	5%	10%	11%	2%
7	2%	2%	9%	2%	7%	21%	100%	11%	2%	2%	7%	5%	11%	5%	7%	8%	2%
8	2%	2%	7%	2%	7%	19%	13%	100%	1%	2%	7%	5%	14%	5%	6%	7%	2%
9	2%	2%	6%	2%	7%	13%	7%	32%	100%	2%	4%	3%	7%	4%	4%	4%	2%
10	3%	2%	7%	3%	7%	34%	5%	6%	2%	100%	4%	3%	6%	5%	5%	6%	2%
11	2%	2%	6%	2%	6%	16%	7%	8%	2%	1%	100%	6%	8%	5%	14%	13%	1%
12	2%	2%	6%	2%	6%	17%	9%	10%	1%	2%	9%	100%	10%	5%	9%	10%	1%
13	2%	2%	6%	2%	6%	19%	10%	10%	1%	2%	9%	8%	100%	4%	8%	10%	2%
14	2%	2%	6%	2%	6%	15%	6%	7%	2%	1%	14%	5%	7%	100%	15%	11%	1%
15	2%	2%	6%	2%	6%	18%	7%	8%	2%	2%	14%	5%	8%	5%	100%	12%	1%
16	2%	2%	6%	2%	6%	19%	9%	9%	2%	2%	10%	8%	10%	5%	8%	100%	1%
17	2%	2%	5%	2%	7%	10%	3%	5%	2%	2%	8%	2%	4%	31%	9%	6%	100%

Además de conseguir una mejor solución que considerando una u otra matriz por separado, se consigue una matriz de impacto que varía con el tiempo. De esta manera obtenemos una matriz que representa fielmente el estado de la red en cada instante de tiempo, es decir, la matriz varía en función de la degradación que presentan los nodos de la red. Esto tiene lógica pues no debería de ser la misma matriz de impacto en el caso de que todos los nodos se degradaran a la vez que en el caso que la degradación de varios nodos fuera mayor a la degradación del resto de nodos de la red.

Una vez que se han obtenido ambas matrices con el programa de *Wolfram Mathematica,* la combinación de las mismas, representada en las ecuaciones (27 , 28), se lleva a cabo de manera sencilla y en tiempo real en un modelo de Excel que representa la red como se puede ver en la Fig. 17.

El tamaño de los nodos representa el número de caminos independientes que llegan a los mismos y el color indica si están por encima de un determinado valor crítico (verde), o por debajo (rojo).

Se realiza un cuadro de control representado en la Fig. 17, mediante el cual se va a introducir la vida útil esperada de los componentes y la inoperabilidad debida a pertubaciones externas, parámetros a partir de los cuales se va a calcular la idegradación de los nodos. Además se introduce el tiempo para el que se quiere conocer el estado de la red.



TIEMPO	2

NODO	TIEMPO DE VIDA ESTIMADO	PERTURBACIÓN EXTERNA
1	10	0
2	10	0
3	10	0
4	10	0
5	10	0
6	10	0
7	10	0
8	10	0
9	10	0
10	10	0
11	10	0
12	10	0
13	10	0
14	10	0
15	10	0
16	10	0
17	10	0

Figura 17 Implementación del modelo en Excel

Los resultados obtenidos se muestran como aparece en la Fig. 18. En primer lugar se muestra la tasa de fallos y la degradación debida a los fallos externos. En segundo lugar se muestran los caminos independientes disponibles que entran a cada nodo y la reducción en procentaje de los mismos debido a la degradación de todos los componentes. Por último se muestra el ranking de riesgo de los nodos, de mayor a menor riesgo.

RESULTADOS

NODO	LAMBDA	PERTURBACIÓN EXTERNA	CAMINOS INDEP. DISPONIBLES	REDUCCIÓN DEL PATH DIVERSITY	RANKING RIESGO
1	0.10	0.00	18.69	36%	10
2	0.10	0.00	16.29	36%	2
3	0.10	0.00	23.34	36%	17
4	0.10	0.00	18.69	36%	9
5	0.10	0.00	23.04	36%	1
6	0.10	0.00	24.20	36%	4
7	0.10	0.00	23.54	36%	14
8	0.10	0.00	22.78	36%	8
9	0.10	0.00	16.81	36%	5
10	0.10	0.00	16.15	36%	11
11	0.10	0.00	23.17	36%	3
12	0.10	0.00	25.09	36%	15
13	0.10	0.00	24.94	36%	7
14	0.10	0.00	22.60	36%	6
15	0.10	0.00	23.45	36%	16
16	0.10	0.00	24.93	36%	13
17	0.10	0.00	16.35	36%	12

Figura 18 Cuadro de resultado obtenidos

3.3 Casos de estudio

Para calcular la inoperabilidad y por tanto el riesgo de cada nodo en los diferentes casos de estudio además de necesitar la matriz S, se require del vector $c(t)$, el cual está constituido por dos términos. En primer lugar la degradación $F(t)$ y en segundo lugar la perturbación debida a fallos externos $e(t)$. Con respecto al segundo término va a depender del caso de estudio y se va a evaluar como el porcentaje de inoperabilidad del nodo afectado, pero en el caso de la degradación se requiere de una función del tiempo que represente de manera adecuada este fenómeno.

Puesto que interesa un mantenimiento centrado en fiabilidad, se va a suponer de aquí en adelante que la degradación de los puntos de la red sigue una distribución exponencial con tasa de fallo λ, por lo que la probabilidad de fallo con el tiempo es:

$$F_i(t) = 1 - e^{-\lambda_i \cdot t} \tag{32}$$

Siendo $1/\lambda$ la vida de servicio o vida útil estimada del componente.

Se van a presentar cuatro casos de estudio diferentes.

3.3.1 Caso de estudio 1

En el primer caso de estudio se supone el caso más simple de todos, no existen perturbaciónes externas y todos los componentes pertenecen al mismo fabricante, por lo que todos los puntos de la red se degradan al mismo ritmo. Esto es una situación ideal que no se da en la realidad pero que sirve como punto de partida para ver cómo va a funcionar el modelo. Se supone una vida útil de los equipos de 10 años, quedando la degradación y por tanto el vector $c(t)$ como:

$$c_i(t) = F_i(t) = 1 - e^{-\frac{t}{10}} = 1 - e^{-0.1 \cdot t}$$

Puesto que no existe ningún nodo que se degrade más rápido que el resto y se ha supuesto que no existen perturbaciones externas, la matriz de impacto introducida en el modelo va a ser únicamente la matriz de impacto normalizada. De esta manera se consegue eliminar la redundancia presente en la matriz de impacto inicial sin incurrir en error alguno.

En la Tabla VIII se muestran los resultados obtenidos tras la simulación del primer caso de estudio. En primer lugar se muestra la inoperabilidad de los nodos en los años 1, 3 y 6. En segundo lugar se muestran los caminos independientes disponibles que llegan a cada nodo y por último el ranking de riesgo en función del número de caminos independientes que le queda a cada nodo.

Tabla VIII

RESULTADOS DEL PRIMER CASO DE ESTUDIO

	INOPERABILIDAD			CAMINOS INDEPENDIENTES DISPONIBLES			RANKING DE RIESGO		
AÑO / NODOS	1	3	6	1	3	6	1	3	6
1	19%	52%	90%	23.7	14.1	2.86	10	10	10
2	19%	52%	90%	20.7	12.3	2.50	2	2	2
3	19%	52%	90%	29.6	17.6	3.57	17	17	17
4	19%	52%	90%	23.7	14.1	2.86	9	9	9
5	19%	52%	90%	29.3	17.4	3.53	1	1	1
6	19%	52%	90%	30.7	18.3	3.71	4	4	4
7	19%	52%	90%	29.9	17.8	3.61	14	14	14
8	19%	52%	90%	28.9	17.2	3.49	8	8	8
9	19%	52%	90%	21.3	12.7	2.57	5	5	5
10	19%	52%	90%	20.5	12.2	2.47	11	11	11
11	19%	52%	90%	29.4	17.5	3.55	3	3	3
12	19%	52%	90%	31.9	19.0	3.84	15	15	15
13	19%	52%	90%	31.7	18.8	3.82	7	7	7
14	19%	52%	90%	28.7	17.1	3.46	6	6	6
15	19%	52%	90%	29.8	17.7	3.59	16	16	16
16	19%	52%	90%	31.7	18.8	3.82	13	13	13
17	19%	52%	90%	20.8	12.4	2.50	12	12	12

Como era de esperar, todos los nodos evolucionan con la misma inoperabilidad a lo largo del tiempo. Esto se debe a que las filas de la matriz de impacto normalizada suman 1, y por tanto al degradarse todos los nodos a la vez $F_i(t)$, se saca la degradación como factor común y multiplica a dicho 1.

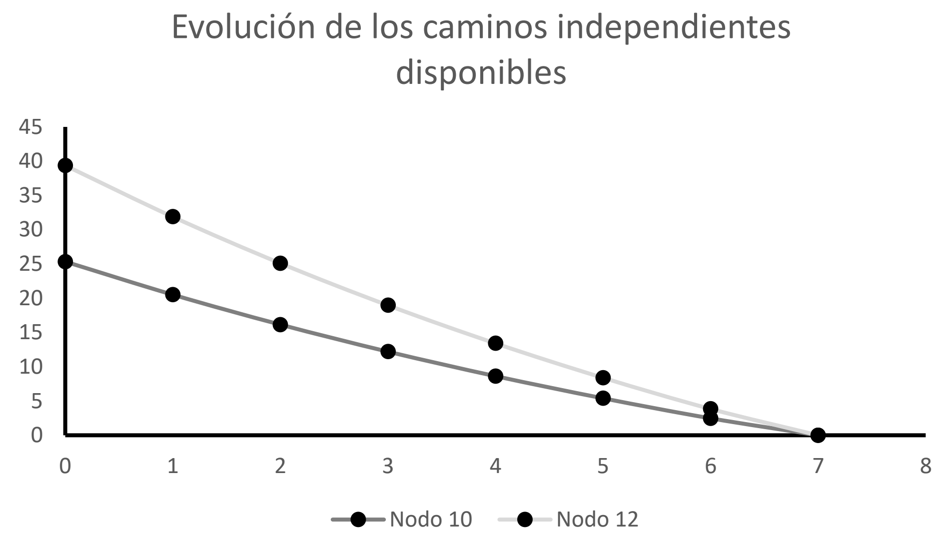
$$x_j = F_i(t) + [F_1(t) \cdot IM_{1j} + F_2(t) \cdot IM_{2j} + \cdots + F_n(t) \cdot IM_{nj}]$$

$$x_j = F_i(t) + \left[F(t) \cdot \left(IM_{1j} + IM_{2j} + \cdots + IM_{nj}\right)\right] = F_i(t) + F(t) \cdot 1 = 2F(t)$$

Por otro lado, aunque la inoperabilidad de los nodos a lo largo del tiempo es la misma, los caminos independientes que llegan a los nodos, no. Esto se traduce en que el Ranking de riesgo no varía en función del tiempo, coincidiendo siempre desde el instante inicial.

Por tanto, el riesgo será mayor en aquellos caminos que en el instante inicial tuvieran menos caminos independientes recibidos, que como se veía anteriormente coinciden con los nodos de los extremos. En contrapartida, los nodos con menos riesgo son aquellos que desde el instante inicial recibían el mayor número de caminos independientes, que en este caso son el 12, el 13 y el 16.

Otra circunstancia a tener en cuenta es que el número de caminos independientes recibidos, y por tanto el riesgo, se va igualando conforme pasa el tiempo. En la gráfica 3 se puede ver la evolución de los caminos independientes disponibles de los nodos 10 y 12 que respectivamente, son el nodo con mayor y con menor riesgo.



Gráfica 3 Evolución de los caminos independientes disponibles

3.3.2 Caso de estudio 2

En el segundo caso de estudio que se presenta, se asume que todos los componentes son de un mismo fabricante y que por tanto se degradan a la vez, pero hay uno que pertenece a otro fabricante y que tiene una vida útil menor. Por otro lado, se va a suponer que no existen perturbaciones externas que afecten a la red.

La degradación de los componentes del primer fabricante, a los que se les supone una vida útil de 10 años al igual que en el caso anterior, sigue la siguiente exponencial.

$$c_i(t) = F_i(t) = 1 - e^{-\frac{t}{10}} = 1 - e^{-0.1 \cdot t}$$

En segundo lugar, el componente 5, que es el que se supone del segundo fabricante, tiene una vida útil de 6 años, y por tanto tiene la siguiente degradación.

$$c_i(t) = F_i(t) = 1 - e^{-\frac{t}{6}} = 1 - e^{-0.17 \cdot t}$$

En la Tabla IX se presentan de la misma manera que en el caso práctico anterior, los resultados obtenidos en este segundo caso de estudio.

Tabla IX

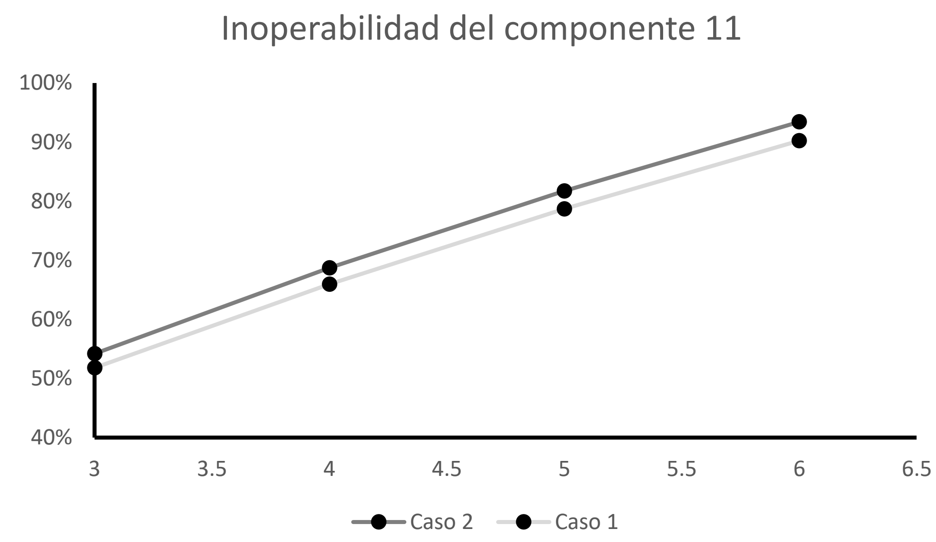
RESULTADOS DEL SEGUNDO CASO DE ESTUDIO

	INOPERABILIDAD			CAMINOS INDEPENDIENTES DISPONIBLES			RANKING DE RIESGO		
AÑO / **NODOS**	1	3	6	1	3	6	1	3	6
1	24%	64%	100%	22.23	10.65	0.00	10	1	1
2	20%	55%	94%	20.39	11.62	1.57	2	4	4
3	20%	55%	94%	29.23	16.69	2.31	17	10	5
4	24%	64%	100%	22.23	10.65	0.00	9	2	2
5	25%	65%	100%	27.16	12.56	0.00	1	17	10
6	20%	54%	93%	30.35	17.40	2.52	4	9	17
7	20%	54%	93%	29.53	16.92	2.44	5	5	9
8	20%	54%	93%	28.57	16.39	2.39	14	14	3
9	20%	55%	94%	21.05	12.02	1.66	8	8	14
10	20%	55%	94%	20.23	11.56	1.61	11	3	8
11	20%	54%	93%	29.08	16.70	2.46	3	11	7
12	20%	54%	93%	31.52	18.14	2.74	15	15	11
13	20%	54%	93%	31.32	18.01	2.69	7	7	15
14	20%	54%	93%	28.35	16.27	2.38	6	6	6
15	20%	54%	93%	29.43	16.89	2.48	16	13	13
16	20%	54%	93%	31.30	18.01	2.71	13	16	16
17	20%	54%	94%	20.49	11.70	1.62	12	12	12

Al contrario que en el caso anterior, los nodos ya no evolucionan todos de la misma manera. En primer lugar y como es lógico, el nodo 5 presenta una inoperabilidad más alta en el tiempo que en el caso anterior, pero además, los nodos de su alrededor también se ven afectados. Los nodos 1 y 4, que únicamente están unidos a la red a través del nodo 5 son los más afectados, mientras que los nodos de la parte superior de la red también se ven afectados en parte.

Atendiendo al ranking de riesgo, comienza de la misma manera que en el caso anterior, pero conforme pasa el tiempo, los nodos 1,4 y 5 van escalando posiciones hasta colocarse en la posición de mayor riesgo. A partir del sexto año, quedan totalmente inoperables. Con respecto al resto de nodos cabe destacar la subida general de los nodos de la parte superior.

Además, la inoperabilidad de todos los nodos ha subido con respecto al primer caso práctico, y mayor es la diferencia cuanto mayor es el tiempo transcurrido y cuanto menor es la vida útil esperada del componente 5. En la gráfica 4 se muestra la evolución del nodo 11, que no tiene contacto directo con 5, pero que debido a la inoperabilidad que este ha difundido a lo largo de la red, se ha visto afectado en mayor medida que en el primer caso de estudio.



Gráfica 4 Inoperabilidad del componente 11 en los casos 1 y 2 de estudio

Se observa en la gráfica 2 un aumento en la inoperabilidad del componente 11 con el tiempo. Se puede concluir que la consecuencia de tener un componente con una vida útil de 6 años, frente al resto de 10 años, es un aumento de la inoperabilidad general de la red, que va creciendo con el tiempo. En el inicio no parecía tener efecto, pero ha ido creciendo, llegando en este caso al sexto año con un 4% más de inoperabilidad en la red en general. Por otra parte, aquellos nodos que se han visto más impactados como el 1 o el 4 a partir del tercer año ya presentaban un 10% más de inoperabilidad que en el primer caso.

3.3.3 Caso de estudio 3

Una vez vistos y analizados los dos casos anteriores se va a suponer un tercer caso en el que la proporción de equipos del primer fabricante, con vida útil de 10 años, con respecto a los del segundo fabricante, de 6 años, es de un 60% distribuidos de manera aleatoria. En la Tabla X se muestra la vida útil esperada de cada uno de los componentes de la red. Además no se suponen fallos en la red debido a perturbaciones externas.

Tabla X

VIDA ÚTIL ESPERADA

NODOS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
VIDA ÚTIL ESPERADA	10	6	6	6	10	10	6	10	10	6	6	10	10	10	6	10	10

Siguiendo la tónica anterior se espera una subida de la inoperabilidad en todos los puntos de la red, aunque en este caso no se puede saber a priori cuales van a ser los nodos que presentan mayor riesgo. En la tabla XI se muestran los resultados y posteriormente se analizan.

Tabla XI

RESULTADOS DEL TERCER CASO DE ESTUDIO

	INOPERABILIDAD			CAMINOS INDEPENDIENTES DISPONIBLES			RANKING DE RIESGO		
AÑO / **NODOS**	1	3	5	1	3	5	1	3	5
1	21%	57%	86%	23.03	14.03	4.16	2	2	2
2	29%	75%	100%	18.10	8.68	0.00	10	10	3
3	27%	71%	100%	26.65	12.65	0.00	17	4	4
4	27%	69%	100%	21.50	10.12	0.00	9	11	7
5	22%	60%	89%	28.03	17.23	4.07	4	15	10
6	22%	59%	88%	29.50	18.40	4.46	1	17	11
7	27%	70%	100%	26.94	12.78	0.00	11	3	15
8	22%	59%	88%	27.79	17.06	4.27	15	7	17
9	21%	57%	86%	20.71	12.51	3.77	3	9	14
10	27%	70%	100%	18.47	8.71	0.00	7	1	9
11	28%	72%	100%	26.31	12.49	0.00	14	14	5
12	22%	59%	88%	30.61	18.72	4.68	8	5	1
13	22%	59%	88%	30.43	18.68	4.69	5	8	8
14	23%	61%	90%	27.28	16.72	3.40	6	6	6
15	28%	71%	100%	26.64	12.65	0.00	16	13	12
16	22%	59%	88%	30.42	18.70	4.70	13	16	13
17	22%	58%	87%	20.05	12.13	3.34	12	12	16

Como se podia prever, los componentes que han caído primero son aquellos que eran del segundo fabricante, que pasado el cuarto año de vida útil quedaron fuera de servicio. A partir de ahí la degradación del resto de los componentes fue mayor que en los casos anteriores.

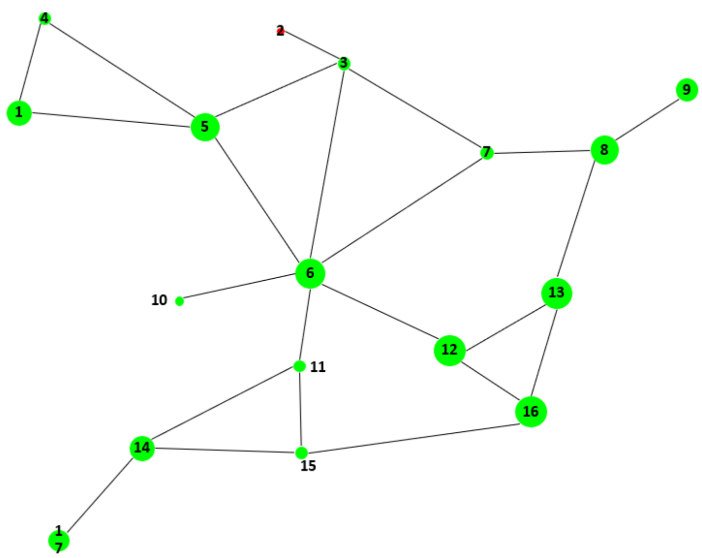


Figura 19 Estado de la red en el cuarto año

En la Fig. 19 se muestra un ejemplo del estado de la red en el cuarto año de funcionamiento, momentos antes de que los nodos del segundo fabricante comiencen a caer. Se puede ver como los nodos 10, 11 y 15 separan a los nodos 14 y 17 de la red, circunstancia que hace que estos dos se coloquen primeros en el ranking de riesgo tran los nodos que han caído, sombreados en gris. Por otro lado se puede ver que la zona fuerte de la red se encuentra en el *cluster* formado por los componentes 12, 13 y 16.

3.3.4 Caso de estudio 4

En este último caso de estudio se va a suponer que todos los componentes son del primer fabricante, teninendo una vída útil de 10 años, pero que en el primer año, el componente 6 sufre un fallo y su capacidad queda reducida al 70%.

La degradación de los componentes del primer fabricante sigue la exponencial que se muestra a continuación.

$$c_i(t) = F_i(t) = 1 - e^{-\frac{t}{10}} = 1 - e^{-0.1 \cdot t}$$

En segundo lugar, el componente 6, que es el que presenta la perturbación externa, tiene la siguiente degradación.

$$c_6(t) = F_i(t) + e_6(t) = 1 - e^{-0.1 \cdot t} + e_6(t)$$

Donde $e_6(t) = 0.7$ a partir de comienzos del primer año.

En la Tabla XII se muestran los resultados tras la simulación del cuarto caso de estudio. La perturbación externa se puede añadir en cualquier instante de tiempo que se desee, lo que lo hace una herramienta muy valiosa en los momentos en los que se requiere una rápida actuación. Por ejemplo, en el caso de estudio actual se querrá saber cuales son los nodos que corren más riesgo mientras el componente 6 se encuentra en reparación. De esta manera se atacarán aquellos nodos que vayan a caer de manera más inmediata.

Tabla XII

RESULTADOS DEL CUARTO CASO DE ESTUDIO

	INOPERABILIDAD			CAMINOS INDEPENDIENTES DISPONIBLES			RANKING DE RIESGO		
AÑO / NODOS	1	2	4	1	2	4	1	3	4
1	19%	61%	90%	23.74	11.50	2.83	10	6	3
2	19%	63%	92%	20.69	9.49	1.94	2	10	5
3	19%	74%	100%	29.64	9.70	0.00	17	3	6
4	19%	61%	90%	23.74	11.50	2.83	9	7	7
5	19%	71%	100%	29.27	10.41	0.00	1	8	8
6	19%	100%	100%	30.73	0.00	0.00	4	5	10
7	19%	73%	100%	29.90	9.87	0.00	14	13	13
8	19%	71%	100%	28.93	10.25	0.00	8	16	16
9	19%	63%	92%	21.35	9.88	2.09	5	15	15
10	19%	100%	100%	20.51	0.00	0.00	11	2	12
11	19%	66%	95%	29.43	12.50	1.76	3	9	11
12	19%	69%	99%	31.87	12.14	0.52	15	12	14
13	19%	71%	100%	31.68	11.34	0.00	7	14	2
14	19%	65%	95%	28.70	12.30	1.83	6	11	9
15	19%	70%	99%	29.78	11.21	0.35	16	1	1
16	19%	71%	100%	31.66	11.46	0.00	13	4	4
17	19%	57%	87%	20.77	10.93	3.34	12	17	17

A la vista de los resultados se pueden sacar varias conclusiones. En primer lugar y como se puede ver en la Tabla V, el componente 6 es el que más impacta sobre la red completa, debido a que es el componente que mayor ganancia de influencia tiene, y por tanto es el punto más crítico de la misma. Estamos por tanto ante el caso más crítico de la red.

Al inicio la evolución seguía el camino del primer caso de estudio, pero después del primer año, cuando el componente 6 falla, la red se degrada drásticamente, llegando al siguiente año con un aumento de las degradaciones de más del 40%. Mientras que en los demás casos de estudio se ha visto que la red suele aguantar entre 5 y 7 años, en este caso a partir del cuarto año ya está completamente inoperable.

Una vez que el nodo 6 falla hay que comenzar a tomar decisiones. En el segundo año ya indicaba que los nodos de mayor riesgo eran el 6 y el 10, que caen en el tercero, y del tercero al cuarto vuelven a caer los 6 siguientes nodos del ranking de riesgo. Por tanto parece buen punto de partida comenzar a centrar la atención en aquellos nodos que ocupan las primeras posiciones del ranking de riesgo.

Otro tema a tratar es el refuerzo de los componentes a la vista de la Tabla V. En dicha tabla se puede ver el impacto de cada nodo sobre el total de la red, es decir, ante un fallo en un nodo, las consecuencias que tiene en la red. Para evitar situaciones como este caso de estudio parece coherente centrarse en vigilar aquellos componentes cuyo impacto sobre la red sea mayor. Se tratará de llevar a cabo mantenimientos preventivos y predictivos en dichos puntos de cara a evitar estos fallos. Por otro lado los componentes de los extremos como el 2, el 9 o el 10, puesto que tienen un bajo impacto sobre la red no merecerá la pena realizar mantenimientos preventivos y predictivos puesto que ante un fallo de los mismos no se van a producir grandes consecuencias. Se deja abierta la línea de investigación de evaluar en que componentes conviene hacer un mantenimiento más exhaustivo a pesar del coste que conlleva para evitar las consecuencias, y en que componentes esto no sale rentable.

Igualmente se ofrece la posibilidad en el modelo de introducir variables como la temperatura, que afecten a la vida útil del componente a lo largo del tiempo en función de la localización del mismo.

Estos cuatro casos de estudio que se han presentado son los cuatro ejemplos más generales que se pueden dar, pero el modelo funciona igualmente para cualquier valor de vida útil de los componentes y cualquier perturbación externa que se quiera tener en cuenta. Además de que los resultados son prometedores, el modelo funciona para todo tipo de redes, lo que lo hace una valiosa herramienta que no conlleva prácticamente gasto ni tiempo.

4 CONCLUSIONES

Una vez concluido el trabajo y mostrado el modelo que se ha desarrollado, se presentan las principales conclusiones obtenidas, así como las posibles líneas de investigación futuras que surgen a raíz del trabajo.

En primer lugar hace frente a la falta de modelos heurísticos para el cálculo de supervivencia de redes. Estos modelos permiten calcular de manera rápida y sencilla los puntos con mayor riesgo de la red. Se basan en modelar cada punto como una entidad única cuya actuación o disponibilidad depende del correcto funcionamiento de los demás puntos. En la Fig. 20 [34] se puede ver el rango de actuación en el que se sitúa el modelo. Se encuentra en la intersección entre los modelos holísticos y los modelos topológicos, puesto que nos hemos basado en la teoría de grafos para poder desarrollar el modelo. De esta manera, partiendo de unos datos sencillos como es la topología, se obtiene una importante información estratégica que ayuda a planificar el mantenimiento y consumir los recursos donde realmente son necesarios.

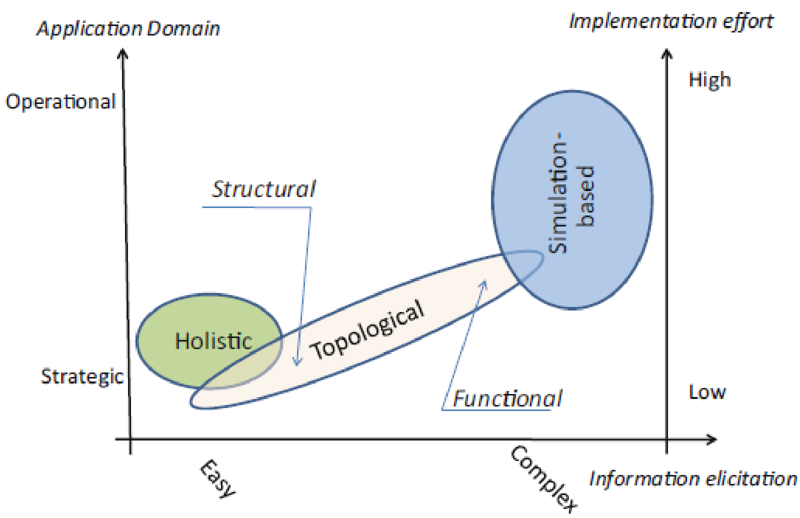


Figura 20 Tipos de modelado de dependencias

Aunque generalmente los modelos holísticos funcionan con datos económicos o empíricos, como datos históricos u opiniones de expertos, para modelar las dependencias entre los distintos puntos, en este trabajo se ha modelado en función de la topología, más concretamente de una novedosa técnica llamada *Path Diversity* o diversidad de caminos. De esta manera se evita la subjetividad, creando un modelo con buena aproximación y con la mayor objetividad posible.

En [25] se concluye que existen multitud de modelos que cubren diferentes aspectos del problema y que combinándolos se pueden eliminar las desventajas que presentan cada uno de ellos. En el modelo presentado se han combinado dos técnicas como son el IIM, que como se ha podido investigar, no encaja del todo bien con redes grandes, y el *Path Diversity*, que está basado en la topología de las redes. Esta combinación permite un amplio uso del modelo.

A la vista de los resultados y del análisis de los casos prácticos se pone de manifiesto que el modelo permite diseñar redes más resilientes mediante el aumento de las capacidades de absorción, anticipación, adaptación y restauración. Esto se logra minimizando los efectos de los fallos promoviendo la diversidad de caminos, evitando o previniendo los fallos tenindo más controlados aquellos puntos que tienen mayor riesgo, adaptándose a los fallos premitiendo un funcionamiento contínuo y recuperándose de manera rápida con equipos de mantenimiento cerca de los puntos críticos de la red.

Gracias a este modelo conseguimos una valiosa herramienta que nos permite con un reducido esfuerzo conseguir una escala de riesgo de todos los puntos de la red. Una vez que se han calculado las matrices de impacto, que es el grueso del modelo, únicamente se implementan en un programa de Excel o Xcelsius, programas fácilmente manejables y entendibles por las organizaciones que les permiten en función de las variables que deseen y prácticamente en tiempo real ver como evoluciona el ranking de riesgo de la red de manera objetiva. Además, ante la caída de uno o varios nodos en la red no hay que volver a realizar una simulación. Únicamente es necesaria una simulación en el caso inicial, lo cual lo hace un modelo muy interesante.

Frente a multitud de modelos de aproximación que se centran en áreas de conocimiento específicas, o en casos particulares, el modelo propuesto funcióna para cualquier tipo de red y cualquier tamaño de la misma. La única variación es el tiempo de compilación para la obtención de las matrices, que aun así es reducido. Una vez calculadas estas matrices no se requiere más esfuerzo. Esto lo hace un modelo polivalente.

El actual modelo presentado está basado en la topogía de la red y en la degradación de los puntos de la misma, no obstante se han abierto nuevas líneas de investigación de cara a la mejora del mismo. Por un lado se está investigando un nuevo modelo en el que no sólo se tenga en cuenta la degradación de los nodos, sino también la degradación de los enlaces. Muchas veces dependiendo del tipo de red, interesará más prestar atención a los enlaces en lugar de los nodos, o viceversa, mientras que en otros casos interesará tener en cuenta ambos.

Otro de los caminos de investigación que se está llevando a cabo es la obtención de las matrices de impacto no sólo basadas en la topología de la red, sino también en la capacidad que pueden acoger los nodos y los enlaces. En este modelo se ha supuesto que cada enlace o cada nodo puede acoger la capacidad que se desee, pero en la realidad esto no suele ser así.

Por último, se está estudiando la adaptación del modelo a los diferentes casos de estudio. Como se puede ver en el desarrollo del modelo, se tenían que combinar la matriz de impacto y la matriz de impacto normalizada en función de la degradación de los nodos de la red. Se está investigando acerca de en función del campo en el que nos encontremos, utilizar unos factores correctores que se adapten mejor a las necesidades, logrando así una mejor aproximación.

REFERENCIAS

[1] J. Sterbenz, E. Çetinkaya, M. Hameed, A. Jabbar, S. Qian and J. Rohrer, "Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation", Telecommunication Systems, 2011.

[2]J. Rohrer, A. Jabbar and J. Sterbenz, "Path diversification: A multipath resilience mechanism", *2009 7th International Workshop on Design of Reliable Communication Networks*, 2009.

[3]S. Soni, R. Gupta and H. Pirkul, "Survivable Network Design: The State of the Art", *Information Systems Frontiers*, vol. 1, no. 3, pp. 303-315, 1999.

[4]J. Roher and J. Sterbenz, "Predicting topology survivability using path diversity", *3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2011.

[5]J. Rohrer, R. Naidu and J. Sterbenz, "Multipath at the transport layer: An end-to-end resilience mechanism", *2009 International Conference on Ultra Modern Telecommunications & Workshops*, 2009.

[6]A. Azni, R. Ahmad, Z. Noh, F. Hazwani and N. Hayaati, "Systematic Review for Network Survivability Analysis in MANETS", *Procedia - Social and Behavioral Sciences*, vol. 195, pp. 1872-1881, 2015.

[7]Y. Cheng, M. Gardner, J. Li, R. May, D. Medhi and J. Sterbenz, "Optimised heuristics for a geodiverse routing protocol", *2014 10th International Conference on the Design of Reliable Communication Networks (DRCN)*, 2014.

[8]A. Zolfaghari and F. Kaudel, "Framework for network survivability performance", *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 1, pp. 46-51, 1994.

[9]K. Steiglitz, P. Weiner and D. Kleitman, "The Design of Minimum-Cost Survivable Networks", *IEEE Transactions on Circuit Theory*, vol. 16, no. 4, pp. 455-460, 1969.

[10]D. Rajan and A. Atamtürk, "Survivable Network Design: Routing of Flows and Slacks", *Operations Research/Computer Science Interfaces Series*, pp. 65-81, 2003.

[11]M. Garg and J. Smith, "Models and algorithms for the design of survivable multicommodity flow networks with general failure scenarios", *Omega*, vol. 36, no. 6, pp. 1057-1071, 2008.

[12]H. Kerivin and A. Mahjoub, "Design of Survivable Networks: A survey", *Networks*, vol. 46, no. 1, pp. 1-21, 2005.

[13]M. Grötschel, C. Monma and M. Stoer, "Chapter 10 Design of survivable networks", *Handbooks in Operations Research and Management Science*, pp. 617-672, 1995.

[14]D. Habibi and Q. Viet, "Graph Theory for Survivability Design in Communication Networks", *New Frontiers in Graph Theory*, 2012.

[15]R. Bhandari, "Optimal physical diversity algorithms and survivable networks", *Proceedings Second IEEE Symposium on Computer and Communications*.

[16]M. Wilson, "The quantitative impact of survivable network architectures on service availability", *IEEE Communications Magazine*, vol. 36, no. 5, pp. 122-126, 1998.

[17]R. Ellison, R. Linger, T. Longstaff and N. Mead, "Survivable network system analysis: a case study", *IEEE*

Software, vol. 16, no. 4, pp. 70-77, 1999.

[18]J. Zhou, N. Huang, X. Sun, K. Wang and H. Yang, "A new model of network cascading failures with dependent nodes", *2015 Annual Reliability and Maintainability Symposium (RAMS)*, 2015.

[19]M. Stoer, *Design of survivable networks*. Berlin: Springer-Verlag, 1992.

[20]P. Heegaard and K. Trivedi, "Network survivability modeling", *Computer Networks*, vol. 53, no. 8, pp. 1215-1234, 2009.

[21]J. Sterbenz, D. Hutchison, E. Çetinkaya, A. Jabbar, J. Rohrer, M. Schöller and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines", *Computer Networks*, vol. 54, no. 8, pp. 1245-1265, 2010.

[22] J. P. Sterbenz and D. Hutchison. (2008, April) ResiliNets: Multilevel resilient and survivable networking initiative. wiki. http://wiki.ittc.ku.edu/resilinets.

[23] Y. Haimes and P. Jiang, "Leontief-Based Model of Risk in Complex Interconnected Infrastructures", *Journal of Infrastructure Systems*, vol. 7, no. 1, pp. 1-12, 2001.

[24]S. Rinaldi, J. Peerenboom and T. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies", IEEE Control Systems Magazine, vol. 21, no. 6, pp. 11-25, 2001.

[25]R. Setola, V. Rosato, E. Kyriakides and E. Rome, Managing the complexity of critical infrastructures.

[26] J. P. Sterbenz and D. Hutchison. (2008, April) ResiliNets: Multilevel resilient and survivable networking initiative wiki. http://wiki.ittc.ku.edu/resilinets. [Online].

[27] Walker, B., Hollinger, C.S., Carpenter, S.R. and Kinzing, A. (2004) "Resilience, Adaptability and Transformability in Social-ecological Systems" Ecology and Society. 9 (2) p.5.

[28] Francis R, Bekera B (2014) A metric and frameworks for resilience analysis of engineered and infrastructure systems. Reliab Eng Syst Saf 121:90–103.

[29] Alsubaie A, Alutaibi K, Marti J (2016) Resilience assessment of interdependent critical infrastructure. In: Rome E, Theocharidou M, Wolthusen S (eds) Critical information infrastructures security, 10th international conference, CRITIS 2015, Berlin, Germany, 5–7 Oct 2015, Revised Selected Papers, pp 43–55

[30] International Electrotechnical Commission (IEC) (2010) Area 191: Quality of service. www.electropedia.org

[31] J. Gómez Fernández and A. Crespo Márquez, Maintenance Management in Network Utilities. London: Springer London, 2012.

[32] J. Gómez Fernández, Mantenimiento de Redes de Servicios.

[33] Setola R, De Porcellinis S (2008) A methodology to estimate input-output inoperability model parameters. In: Critical information infrastructures security 2007. Lecture Notes in Computer Science. Springer, Berlin, pp 149–160

[34] Setola R (2010) How to measure the degree of interdependencies among critical infrastructures. Int J Syst Syst Eng 2(1):38–59

[35] R. Joana Oliveira, "El ataque de 'ransomware' se extiende a escala global", EL PAÍS, 2017. [Online]. https://elpais.com/tecnologia/2017/05/12/actualidad/1494586960_025438.html.

[36] J. Moubray, Reliability-centered maintenance. New York: Industrial Press, 2001.

ANEXO I. CÓDIGOS

1. Código para el cálculo de las matrices de impacto.

Se presenta a continuación el código programado en *Wolfram Mathematica* que corresponde con la programación del modelo propuesto.

```
g=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g];

EdgeList[g]//MatrixForm;

Enlaces=EdgeList[g];

CapacidadNodo={15,10,20,15,25,35,25,20,10,10,20,15,20,18,15,15,10};

CapacidadesEnlaces={35,35,35,45,50,45,35,40,40,40,40,40,40,30,35,35,35,35,35,35,35,35,35};

VertexList[g];

nnodos=17;

TableForm[Normal@AdjacencyMatrix[g],TableHeadings->{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<nnodos+1,i=i+1,

For[j=1,j<nnodos+1,j=j+1,

If[i!=j,

mat=AppendTo[L,FindPath[g,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g,i,j,Infinity]];

];

];

];



PD=Table[0,nnodos,nnodos];

PC=Table[0,nnodos,nnodos];

For[q=1,q<Length[mat]+1,q=q+1,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];


];

];


DiversityMatrix[[j,i]]=1-(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]],mat[[q,i]]]]-2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;

DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];

];

];


DiversityMatrix2=Table[0,Length[mat2[[q]]],Length[mat[[q]]]];
```

```
For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
```

```
If[i==P[[1,x]],
cabezadegrupo=1;

];
];



For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;


If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
```

```
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
```

```
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[ mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]] || mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnlaces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";

For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;

];
];

"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
```

```
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];


If[PCT2== 0 ,

PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];

PC[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;

];


];


For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1,

PD[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;

Capacidad3={};

For[r=2,r<Length[mat[[q,1]]],r=r+1,

Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];

];

For[r=1,r<Length[mat[[q,1]]],r=r+1,

For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,

If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];

];

];

];

PC[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];

];


N[PD//MatrixForm]

N[PC//MatrixForm]


g1=Graph[{2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g1];

nnodos=VertexCount[g1];

TableForm[Normal@AdjacencyMatrix[g1],TableHeadings->{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=1&&j!=1,

mat=AppendTo[L,FindPath[g1,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g1,i,j,Infinity]];

];

If[i!=j&&(i== 1||j== 1),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];

];

];


PD1=Table[0,17,17];

PC1=Table[0,17,17];


For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]>0,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;
```

```
For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];



For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];



A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
```

```
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,
If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
```

```
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD1[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[mat[[q,P[[j,i]]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]] || mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnlaces[[z]]];


];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];


];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
```

```
];


];
If[Caminos[[r]]>1,


CapacidadPico[[r]]=CapacidadPico[[r]]-Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];



"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal1[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];


If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];


PC1[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;
];


];
];
For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD1[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC1[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
];


N[PD1//MatrixForm]
N[PC1//MatrixForm]



g2=Graph[{1<->4,1<->5,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]
VertexList[g2];
nnodos=VertexCount[g2]
TableForm[Normal@AdjacencyMatrix[g2],TableHeadings->{c=Style[#,Blue]&/@%,c}];


NFilas=0;
```

```
L={};
M={};
For[i=1,i<17+1,i=i+1,
For[j=1,j<17+1,j=j+1,
If[i!=j&&i!=2&&j!=2,
mat=AppendTo[L,FindPath[g2,i,j,Infinity,All]];
mat2=AppendTo[M,FindVertexIndependentPaths[g2,i,j,Infinit
y]];
];
If[i!=j&&(i== 2||j== 2),
mat=AppendTo[L,0];
mat2=AppendTo[M,0];
];
];
];

PD2=Table[0,17,17];
PC2=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];
DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
```

```
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];


A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];


If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];


For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;


If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];


];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];


];
];
];
];
```

```
P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];
NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD2[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]] || mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnlaces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
```

```
];


];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];


];
If[Caminos[[r]]>1,


CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];


"Sería como el porcentaje en el que se redunda la capacidad del
nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]]+(PCT2)*(matoriginal2[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q
,1]]]]]]];
];


If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]];
];


PC2[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbas
e;
];


];
];








For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD2[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,
1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]]          ||
mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]]
;
];
];
```

```
];
PC2[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[
Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]
]]]];
];
];

N[PD2//MatrixForm]
N[PC2//MatrixForm]

g3=Graph[{1<->4,1<->5,4<->5,5<->6,6<->7,6<->10,6<-
>11,6<->12,7<->8,8<->9,8<->13,11<->14,11<->15,12<-
>13,12<->16,13<->16,14<->15,14<->17,15<-
>16},VertexShapeFunction->"Name",VertexStyle->Blue]
VertexList[g3];
nnodos=VertexCount[g3];
TableForm[Normal@AdjacencyMatrix[g3],TableHeadings-
>{c=Style[#,Blue]&/@%,c}];

NFilas=0;

L={};
M={};
For[i=1,i<17+1,i=i+1,
For[j=1,j<17+1,j=j+1,
If[i!=j&&i!=2&&j!=2&&i!=3&&j!=3,
mat=AppendTo[L,FindPath[g3,i,j,Infinity,All]];
mat2=AppendTo[M,FindVertexIndependentPaths[g3,i,j,Infinit
y]];
];
If[i!=j&&(i== 2||j== 2||i== 3||j== 3),
mat=AppendTo[L,0];
mat2=AppendTo[M,0];
];
];
];

PD3=Table[0,17,17];
PC3=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
```

```
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
```

```
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];


If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD3[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
```

```
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";

For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];

"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]]+(PCT2)*(matoriginal3[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q
,1]]]]]]];
];

If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]];
];

PC3[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbas
e;
];
```

```
];
];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD3[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC3[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
];

N[PD3//MatrixForm]
N[PC3//MatrixForm]

g4=Graph[{1<->5,2<->3,3<->5,3<->6,3<->7,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]
VertexList[g4];
nnodos=VertexCount[g4];
TableForm[Normal@AdjacencyMatrix[g4],TableHeadings->{c=Style[#,Blue]&/@%,c}];

NFilas=0;

L={};
M={};
For[i=1,i<17+1,i=i+1,
For[j=1,j<17+1,j=j+1,
If[i!=j&&i!=4&&j!=4,
mat=AppendTo[L,FindPath[g4,i,j,Infinity,All]];
mat2=AppendTo[M,FindVertexIndependentPaths[g4,i,j,Infinity]];
];
If[i!=j&&(i== 4||j== 4),
mat=AppendTo[L,0];
mat2=AppendTo[M,0];
];
];
];

PD4=Table[0,17,17];
PC4=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
```

```
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
```

```
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
```

```
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD4[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnlaces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
```

```
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";

For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];

"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal4[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

PC4[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;
];

];
];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD4[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC4[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
];

N[PD4//MatrixForm]
N[PC4//MatrixForm]

g5=Graph[{1<->4,2<->3,3<->6,3<->7,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]
VertexList[g5];
nnodos=VertexCount[g5];
TableForm[Normal@AdjacencyMatrix[g5],TableHeadings->{c=Style[#,Blue]&/@%,c}];

NFilas=0;

L={};
M={};
For[i=1,i<17+1,i=i+1,
For[j=1,j<17+1,j=j+1,
```

```
If[i!=j&&i!=5&&j!=5,
mat=AppendTo[L,FindPath[g5,i,j,Infinity,All]];
mat2=AppendTo[M,FindVertexIndependentPaths[g5,i,j,Infinit
y]];
];
If[i!=j&&(i== 5||j== 5),
mat=AppendTo[L,0];
mat2=AppendTo[M,0];
];
];
];


PD5=Table[0,17,17];
PC5=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
```

```
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;
For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
```

```
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD5[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]] || mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnlaces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
```

```
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";

For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];


"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
```

```
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal5[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

PC5[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;
];

];
];


For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD5[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC5[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
```

```
];

N[PD5//MatrixForm]
N[PC5//MatrixForm]

g6=Graph[{1<->4,1<->5,2<->3,3<->5,3<->7,4<->5,7<->8,8<->9,8<->13,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]
VertexList[g6];
nnodos=VertexCount[g6];
TableForm[Normal@AdjacencyMatrix[g6],TableHeadings->{c=Style[#,Blue]&/@%,c}];

NFilas=0;

L={};
M={};
For[i=1,i<17+1,i=i+1,
For[j=1,j<17+1,j=j+1,
If[i!=j&&i!=6&&j!=6&&i!=10&&j!=10,
mat=AppendTo[L,FindPath[g6,i,j,Infinity,All]];
mat2=AppendTo[M,FindVertexIndependentPaths[g6,i,j,Infinity]];
];
If[i!=j&&(i== 6||j== 6||i== 10||j== 10),
mat=AppendTo[L,0];
mat2=AppendTo[M,0];
];
];
];

PD6=Table[0,17,17];
PC6=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]],mat[[q,i]]]]-2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
```

```
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];


A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];


If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];


For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];
```

```
For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;


If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];
Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
```

```
PD6[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;


Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,


If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],


CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];


];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];


];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];


];
If[Caminos[[r]]>1,


CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];



"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1
]]]]]]]+(PCT2)*(matoriginal6[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q
,1]]]]]]];
];
```

```
If[PCT2== 0,

PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]];

];

PC6[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbas
e;

];

];

];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,

PD6[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;

Capacidad3={};

For[r=2,r<Length[mat[[q,1]]],r=r+1,

Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,
1]][[r]]]]];

];

For[r=1,r<Length[mat[[q,1]]],r=r+1,

For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,

If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]]            ||
mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]]
;

];

];

];

PC6[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[
Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]
]]]];

];

];

N[PD6//MatrixForm]

N[PC6//MatrixForm]

g7=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,4<->5,5<-
>6,6<->10,6<->11,6<->12,8<->9,8<->13,11<->14,11<-
>15,12<->13,12<->16,13<->16,14<->15,14<->17,15<-
>16},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g7];

nnodos=VertexCount[g7];

TableForm[Normal@AdjacencyMatrix[g7],TableHeadings-
>{c=Style[#,Blue]&/@%,c}];

NFilas=0;

L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=7&&j!=7,

mat=AppendTo[L,FindPath[g7,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g7,i,j,Infinit
y]];

];

If[i!=j&&(i== 7||j== 7),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];

];

];

PD7=Table[0,17,17];

PC7=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]>0,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,
```

```
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
```

```
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];
];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
```

```
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD7[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,
```

```
CapacidadPico[[r]]=CapacidadPico[[r]]-Max[Capacidad4[[;;,r]]];

CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);

];

PCT=PCT+Max[Capacidad4[[;;,r]]];

];

"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,

If[Caminos[[r]]>1,

PCT2=PCT2+CapacidadPico[[r]]/Grupos;

];

];


"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";

If[PCT2>0,

PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal7[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];


If[PCT2== 0,

PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];


PC7[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;

];


];

];


For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,

PD7[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;

Capacidad3={};
```

```
For[r=2,r<Length[mat[[q,1]]],r=r+1,

Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];

];

For[r=1,r<Length[mat[[q,1]]],r=r+1,

For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,

If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];

];

];

];

PC7[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];

];


N[PD7//MatrixForm]

N[PC7//MatrixForm]


g8=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g8];

nnodos=VertexCount[g8];

TableForm[Normal@AdjacencyMatrix[g8],TableHeadings->{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=8&&j!=8&&i!=9&&j!=9,

mat=AppendTo[L,FindPath[g8,i,j,Infinity,All]];
```

```
mat2=AppendTo[M,FindVertexIndependentPaths[g8,i,j,Infinit
y]];

];

If[i!=j&&(i== 8||j== 8||i== 9||j== 9),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];

];

];

PD8=Table[0,17,17];

PC8=Table[0,17,17];


For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]>0,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];

];

];


DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;

DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];

];

];


DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,

For[i=1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat2[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];

];

];


DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);

];

];



For[j=1,j<Length[mat2[[q]]]+1,j=j+1,

salir=0;

For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,

If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],

If[DiversityMatrix2[[j,i]]<1,

candidato=1;

For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,

If[k!= j,

If[DiversityMatrix2[[k,i]]== 1,

candidato=1;

];

If[DiversityMatrix2[[k,i]]<1,

candidato=0;

];

];
```

```
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&& cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;
];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
```

```
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD8[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];


];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];


];
];
];
PCT=0;
PCT2=0;
Grupos=0;
```

```
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];


];
If[Caminos[[r]]>1,


CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];




"Sería como el porcentaje en el que se redunda la capacidad del
nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]]+(PCT2)*(matoriginal8[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q
,1]]]]]]];
];


If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]];
];


PC8[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbas
e;
];


];
];








For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD8[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,
1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]]            ||
mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]]
;
];
];
];
PC8[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[
Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]
]]]];
];
];


N[PD8//MatrixForm]
N[PC8//MatrixForm]
```

```
g9=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->13,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g9];

nnodos=VertexCount[g9];

TableForm[Normal@AdjacencyMatrix[g9],TableHeadings->{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=9&&j!=9,

mat=AppendTo[L,FindPath[g9,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g9,i,j,Infinity]];

];

If[i!=j&&(i== 9||j== 9),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];

];

];

PD9=Table[0,17,17];

PC9=Table[0,17,17];


For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]>0,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];

];

];


DiversityMatrix[[j,i]]=1-(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]],mat[[q,i]]]]-2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;

DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];

];

];


DiversityMatrix2=Table[0,Length[mat2[[q]]],Length[mat[[q]]]];



For[j=1,j<Length[mat2[[q]]]+1,j=j+1,

For[i=1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat2[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}=={mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}=={mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];

];
```

```
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
```

```
P[[j-1,k]]=0;
];



];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD9[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
```

```
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]]][[r]]<->mat[[q,P[[j,i]]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]]][[r+1]]<->mat[[q,P[[j,i]]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];


];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];


];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];
];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];



"Sería como el porcentaje en el que se redunda la capacidad del
nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]]+(PCT2)*(matoriginal9[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Leng
th[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q
,1]]]]]]];
];


If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]];
];


PC9[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbas
e;
];


];
];






For[q=1,q<Length[mat]+1,q=q+1,
```

```
If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,

PD9[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;

Capacidad3={};

For[r=2,r<Length[mat[[q,1]]],r=r+1,

Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,
1]][[r]]]]];

];

For[r=1,r<Length[mat[[q,1]]],r=r+1,

For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,

If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]]            ||
mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]]
;

];

];

];

PC9[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[
Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]
]]]];

];

];


N[PD9//MatrixForm]

N[PC9//MatrixForm]


g10=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<-
>5,5<->6,6<->7,6<->11,6<->12,7<->8,8<->9,8<->13,11<-
>14,11<->15,12<->13,12<->16,13<->16,14<->15,14<-
>17,15<->16},VertexShapeFunction->"Name",VertexStyle-
>Blue]

VertexList[g10];

nnodos=VertexCount[g10];

TableForm[Normal@AdjacencyMatrix[g10],TableHeadings-
>{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=10&&j!=10,

mat=AppendTo[L,FindPath[g10,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g10,i,j,Infin
ity]];

];

If[i!=j&&(i== 10||j== 10),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];

];

];

PD10=Table[0,17,17];

PC10=Table[0,17,17];


For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]>0,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},
```

```
ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];
```

```
If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];


For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;


];
];


For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;


If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];
];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];
```

```
If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD10[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT
;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de
origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],
CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
```

```
];

"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,

If[Caminos[[r]]>1,

PCT2=PCT2+CapacidadPico[[r]]/Grupos;

];

];




"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";

If[PCT2>0,

PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal10[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];


If[PCT2== 0,

PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];


PC10[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;

];


];

];




For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,

PD10[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;

Capacidad3={};

For[r=2,r<Length[mat[[q,1]]],r=r+1,

Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];

];

For[r=1,r<Length[mat[[q,1]]],r=r+1,

For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,

If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];

];

];

];

PC10[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];

];

N[PD10//MatrixForm]

N[PC10//MatrixForm]




g11=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->12,7<->8,8<->9,8<->13,12<->13,12<->16,13<->16,14<->15,14<->17,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g11];

nnodos=VertexCount[g11];

TableForm[Normal@AdjacencyMatrix[g11],TableHeadings->{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=11&&j!=11,

mat=AppendTo[L,FindPath[g11,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g11,i,j,Infinity]];

];

If[i!=j&&(i== 11||j== 11),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];
```

```
];
];
PD11=Table[0,17,17];
PC11=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];


For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
```

```
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;
If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
```

```
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD11[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT
;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
```

```
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";

For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];

"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal11[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
PC11[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;
];

];
];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD11[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC11[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
];

N[PD11//MatrixForm]
N[PC11//MatrixForm]

g12=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<-
```

```
>5,5<->6,6<->7,6<->10,6<->11,7<->8,8<->9,8<->13,11<-
>14,11<->15,13<->16,14<->15,14<->17,15<-
>16},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g12];

nnodos=VertexCount[g12];

TableForm[Normal@AdjacencyMatrix[g12],TableHeadings-
>{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=12&&j!=12,

mat=AppendTo[L,FindPath[g12,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g12,i,j,Infin
ity]];

];

If[i!=j&&(i== 12||j== 12),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];

];

];

PD12=Table[0,17,17];

PC12=Table[0,17,17];


For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]>0,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;

];

];

];


DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;

DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];

];

];


DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];



For[j=1,j<Length[mat2[[q]]]+1,j=j+1,

For[i=1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat2[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];

];

];


DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);

];

];



For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
```

```
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];
If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];


For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;


];
];



For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;


If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];
```

```
];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];
If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD12[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT
;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],
```

```
CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];


];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];


];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];


];
If[Caminos[[r]]>1,


CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];



"Sería como el porcentaje en el que se redunda la capacidad del
nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]]+(PCT2)*(matoriginal12[[mat[[q,1]][[1]],mat[[q,1]][[Len
gth[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Len
gth[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[
q,1]]]]]]];
];


If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]];
];


PC12[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCb
ase;
];


];
];


For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD12[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,
1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
```

```
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,

If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]]   ||
mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]]
;

];

];

];

PC12[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min
[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]
]]]]];

];

];


N[PD12//MatrixForm]

N[PC12//MatrixForm]


g13=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<-
>5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,11<-
>14,11<->15,12<->16,14<->15,14<->17,15<-
>16},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g13];

nnodos=VertexCount[g13];

TableForm[Normal@AdjacencyMatrix[g13],TableHeadings-
>{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=13&&j!=13,

mat=AppendTo[L,FindPath[g13,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g13,i,j,Infin
ity]];

];

If[i!=j&&(i== 13||j== 13),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];

];

];

PD13=Table[0,17,17];

PC13=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]>0,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];

];

];


DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;

DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];

];

];


DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];



For[j=1,j<Length[mat2[[q]]]+1,j=j+1,

For[i=1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat2[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},
```

```
ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
```

```
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD13[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT
;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,
```

```
If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]] || mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnlaces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";

For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];

"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal13[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

PC13[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;
];

];
```

```
];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD13[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC13[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
];

N[PD13//MatrixForm]
N[PC13//MatrixForm]

g14=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->15,12<->13,12<->16,13<->16,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]
VertexList[g14];
nnodos=VertexCount[g14];
TableForm[Normal@AdjacencyMatrix[g14],TableHeadings->{c=Style[#,Blue]&/@%,c}];

NFilas=0;

L={};
M={};
For[i=1,i<17+1,i=i+1,
For[j=1,j<17+1,j=j+1,
If[i!=j&&i!=14&&j!=14&&i!=17&&j!=17,
mat=AppendTo[L,FindPath[g14,i,j,Infinity,All]];
mat2=AppendTo[M,FindVertexIndependentPaths[g14,i,j,Infinity]];
];
If[i!=j&&(i== 14||j== 14||i== 17||j== 17),
mat=AppendTo[L,0];
mat2=AppendTo[M,0];
];
];
];
PD14=Table[0,17,17];
PC14=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h+1]],mat[[q,i]][[h]]},
```

```
ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]],mat[[q,i]]]]-2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length [mat[[q]]]];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}=={mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}=={mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j]],mat[[q,i]]]]-2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];
```

```
If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];
];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];
```

```
If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD14[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT
;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],
CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
```

```
];

"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,

If[Caminos[[r]]>1,

PCT2=PCT2+CapacidadPico[[r]]/Grupos;

];

];



"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";

If[PCT2>0,

PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal14[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];


If[PCT2== 0,

PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];


PC14[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;

];


];

];



For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,

PD14[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;

Capacidad3={};

For[r=2,r<Length[mat[[q,1]]],r=r+1,

Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];

];

For[r=1,r<Length[mat[[q,1]]],r=r+1,

For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,

If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],


Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]]
;

];

];

];

PC14[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];

];

];


N[PD14//MatrixForm]

N[PC14//MatrixForm]



g15=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->14,12<->13,12<->16,13<->16,14<->17},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g15];

nnodos=VertexCount[g15];

TableForm[Normal@AdjacencyMatrix[g15],TableHeadings->{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=15&&j!=15,

mat=AppendTo[L,FindPath[g15,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g15,i,j,Infinity]];

];

If[i!=j&&(i== 15||j== 15),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];
```

```
];
];
];
PD15=Table[0,17,17];
PC15=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]
],mat[[q,i]]]]-
2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length
[mat[[q]]]];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;
For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
```

```
];
];
];


A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];



If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];



For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&& cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;


];
];



For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;


If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];



];
];
];
];


P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];


Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
```

```
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD15[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT
;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de
origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
```

```
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";

For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];

"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]]+(PCT2)*(matoriginal15[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

PC15[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;
];

];
];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD15[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC15[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
];

N[PD15//MatrixForm]
N[PC15//MatrixForm]
```

```
g16=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->14,11<->15,12<->13,14<->15,14<->17},VertexShapeFunction->"Name",VertexStyle->Blue]

VertexList[g16];

nnodos=VertexCount[g16];

TableForm[Normal@AdjacencyMatrix[g16],TableHeadings->{c=Style[#,Blue]&/@%,c}];


NFilas=0;


L={};

M={};

For[i=1,i<17+1,i=i+1,

For[j=1,j<17+1,j=j+1,

If[i!=j&&i!=16&&j!=16,

mat=AppendTo[L,FindPath[g16,i,j,Infinity,All]];

mat2=AppendTo[M,FindVertexIndependentPaths[g16,i,j,Infinity]];

];

If[i!=j&&(i== 16||j== 16),

mat=AppendTo[L,0];

mat2=AppendTo[M,0];

];

];

];

PD16=Table[0,17,17];

PC16=Table[0,17,17];


For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]>0,

DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];

If[Length[mat[[q]]]>1,

For[j=1,j<Length[mat[[q]]],j=j+1,

For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}=={mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];

];

];


DiversityMatrix[[j,i]]=1-(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]],mat[[q,i]]]]-2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;

DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];

];

];


DiversityMatrix2=Table[0,Length[mat2[[q]]],Length[mat[[q]]]];



For[j=1,j<Length[mat2[[q]]]+1,j=j+1,

For[i=1,i<Length[mat[[q]]]+1,i=i+1,

ArcosComunes=0;


For[k=1,k<Length[mat2[[q,j]]],k=k+1,

For[h=1,h<Length[mat[[q,i]]],h=h+1,


If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}=={mat[[q,i]][[h]],mat[[q,i]][[h+1]]},


ArcosComunes=ArcosComunes+1;

];

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}=={mat[[q,i]][[h+1]],mat[[q,i]][[h]]},


ArcosComunes=ArcosComunes+1;

];

];

];


DiversityMatrix2[[j,i]]=1-(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j]],mat[[q,i]]]]-2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);

];

];
```

```
For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
```

```
P[[j-1,k]]=0;
];



];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];

PDT=PDA+PDB;
PD16[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT
;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
```

```
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],


CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];


];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];


];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];


];
If[Caminos[[r]]>1,


CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";


For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];



"Sería como el porcentaje en el que se redunda la capacidad del
nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]]+(PCT2)*(matoriginal16[[mat[[q,1]][[1]],mat[[q,1]][[Len
gth[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Len
gth[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[
q,1]]]]]]];
];


If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]];
];


PC16[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCb
ase;
];


];
];


For[q=1,q<Length[mat]+1,q=q+1,


If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD16[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,
1]][[r]]]]];
```

```
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC16[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
];

N[PD16//MatrixForm]
N[PC16//MatrixForm]
```

```
g17=Graph[{1<->4,1<->5,2<->3,3<->5,3<->6,3<->7,4<->5,5<->6,6<->7,6<->10,6<->11,6<->12,7<->8,8<->9,8<->13,11<->14,11<->15,12<->13,12<->16,13<->16,14<->15,15<->16},VertexShapeFunction->"Name",VertexStyle->Blue]
VertexList[g17];
nnodos=VertexCount[g17];
TableForm[Normal@AdjacencyMatrix[g17],TableHeadings->{c=Style[#,Blue]&/@%,c}];

NFilas=0;

L={};
M={};
For[i=1,i<17+1,i=i+1,
For[j=1,j<17+1,j=j+1,
If[i!=j&&i!=17&&j!=17,
mat=AppendTo[L,FindPath[g17,i,j,Infinity,All]];
mat2=AppendTo[M,FindVertexIndependentPaths[g17,i,j,Infinity]];
];
If[i!=j&&(i== 17||j== 17),
mat=AppendTo[L,0];
mat2=AppendTo[M,0];
];
];
];
PD17=Table[0,17,17];
PC17=Table[0,17,17];

For[q=1,q<Length[mat]+1,q=q+1,
If[Length[mat[[q]]]>0,
DiversityMatrix=Table[0,Length[mat[[q]]],Length[mat[[q]]]];
If[Length[mat[[q]]]>1,
For[j=1,j<Length[mat[[q]]],j=j+1,
For[i=j+1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}== {mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat[[q,j]][[k]],mat[[q,j]][[k+1]]}== {mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix[[j,i]]=1-(ArcosComunes+Length[LongestCommonSequence[mat[[q,j]],mat[[q,i]]]]-2)/(2*(Min[Length[mat[[q,j]]],Length[mat[[q,i]]]])-3)1;
DiversityMatrix[[i,j]]=DiversityMatrix[[j,i]];
];
];

DiversityMatrix2=Table[0,Length[mat2[[q]]],Length[mat[[q]]]];
```

```
For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
For[i=1,i<Length[mat[[q]]]+1,i=i+1,
ArcosComunes=0;

For[k=1,k<Length[mat2[[q,j]]],k=k+1,
For[h=1,h<Length[mat[[q,i]]],h=h+1,

If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h]],mat[[q,i]][[h+1]]},

ArcosComunes=ArcosComunes+1;
];
If[{mat2[[q,j]][[k]],mat2[[q,j]][[k+1]]}==
{mat[[q,i]][[h+1]],mat[[q,i]][[h]]},

ArcosComunes=ArcosComunes+1;
];
];
];

DiversityMatrix2[[j,i]]=1-
(ArcosComunes+Length[LongestCommonSequence[mat2[[q,j
]],mat[[q,i]]]]-
2)/(2*(Min[Length[mat2[[q,j]]],Length[mat[[q,i]]]])-3);
];
];

For[j=1,j<Length[mat2[[q]]]+1,j=j+1,
salir=0;
For[i=1,i<Length[mat[[q]]]+1&&salir==0,i=i+1,
If[Length[mat[[q,i]]]<Length[mat2[[q,j]]],
If[DiversityMatrix2[[j,i]]<1,
candidato=1;
For[k=1,k<Length[mat2[[q]]]+1&&candidato==1,k=k+1,
If[k!= j,
If[DiversityMatrix2[[k,i]]== 1,
candidato=1;
];
If[DiversityMatrix2[[k,i]]<1,
candidato=0;
];
];
];
If[candidato== 1,
mat2[[q,j]]=mat[[q,i]];
salir=1;
];
];
];
];
];

A=DiversityMatrix;
A//MatrixForm;
P=Table[0,Length[A],Length[A]];

Colocados={};
For[x=1,x<Length[mat2[[q]]]+1,x=x+1,
bandera2=0;
If[Length[mat2[[q]]]>1,
For[i=1,i<Length[mat[[q]]]+1&& bandera2== 0,i=i+1,
If[mat[[q,i]]== mat2[[q,x]],
P[[1,x]]=i;
Colocados=AppendTo[Colocados,i];
bandera2=1;
];
];
];
];

If[Length[mat2[[q]]]== 1,
P[[1,1]]=1;
Colocados=AppendTo[Colocados,1];
];

For[i=1,i<Length[A]+1,i=i+1,
cabezadegrupo=0;

For[x=1,x<Length[mat2[[q]]]+1&&
cabezadegrupo==0,x=x+1,
If[i==P[[1,x]],
cabezadegrupo=1;

];
```

```
];

For[k=1,k<Length[A]+1&&cabezadegrupo== 0 ,k=k+1,
b=0;

If[0<A[[i,P[[1,k]]]]<1,
For[j=1,j<Length[A]+1 && b==0,j=j+1,
If[P[[j,k]]==0,
P[[j,k]]=i;
b=1;
];
];
For[a=k+1,a<Length[mat2[[q]]]+1,a=a+1,

If[A[[i,P[[1,k]]]]>= A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
For[a=k,a>0,a=a-1,

If[A[[i,P[[1,k]]]]> A[[i,P[[1,a]]]],
P[[j-1,k]]=0;
];

];
];
];
];

P//MatrixForm;
PDA=0;
PDB=0;
NCamT=0;
"Una vez tenemos los caminos divididos en grupos vamos a calcular la Diversidad total entre ambos nodos"
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
PDA=PDA+1;
];
];

Cuenta=Table[0,Length[A]];
NCam=Table[0,Length[A]];
For[i=1,i<Length[A]+1,i=i+1,
If[P[[1,i]]!=0,
Caminos2=0;
For[r=1,r<Length[P]+1,r=r+1,
If[P[[r,i]]!=0,
Caminos2=Caminos2+1;
];
];
];
If[P[[1,i]]!=0,
AGrupo={};
];
For[k=1,k<Length[A],k=k+1,
For[l=k+1,l<Length[A]+1,l=l+1,

If[P[[l,i]]!=0,
Cuenta[[i]]=Cuenta[[i]]+A[[(P[[k,i]]),(P[[l,i]])]];
AGrupo=AppendTo[AGrupo,A[[(P[[k,i]]),(P[[l,i]])]]];
NCam[[i]]=NCam[[i]]+1;
];
];
];

If[NCam[[i]]==0,
If[P[[1,i]]!=0,
NCam[[i]]=NCam[[i]]+1;
];
];

NCamT=NCamT+NCam[[i]];

If[NCam[[i]]>1,
PDB=PDB+(Max[AGrupo])/(PDA);
];
If[NCam[[i]]==1,
PDB=PDB+Cuenta[[i]]/(PDA);
];
];
```

```
PDT=PDA+PDB;
PD17[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PDT
;

Capacidad=Table[0,Length[P],Length[P]];
Capacidad2=Table[0,Length[P],Length[P]];
Capacidad4=Table[0,Length[P],Length[P]];
CapacidadPico=Table[0,Length[P]];
For[i=1,i<Length[P]+1,i=i+1,
If[P[[1,i]]!=0,
For[j=1,j<Length[P]+1,j=j+1,

If[P[[j,i]]!=0,
CapacidadAux={};
CapacidadAux2={};
"Empezamos en 2 para no meter la capacidad del nodo de origen, que no cuenta al igual que en el PD";
For[r=2,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
CapacidadAux=AppendTo[CapacidadAux,CapacidadNodo[[
mat[[q,P[[j,i]]]][[r]]]]];
];
If[Length[mat[[q,P[[j,i]]]]]!=2,
Capacidad[[j,i]]=Min[CapacidadAux];
];
For[r=1,r<Length[mat[[q,P[[j,i]]]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,P[[j,i]]]][[r]]<->mat[[q,P[[j,i]]]][[r+1]]==Enlaces[[z]]
|| mat[[q,P[[j,i]]]][[r+1]]<->mat[[q,P[[j,i]]]][[r]]==Enlaces[[z]],

CapacidadAux2=AppendTo[CapacidadAux2,CapacidadesEnl
aces[[z]]];

];
];
];
Capacidad2[[j,i]]=Min[CapacidadAux2];
];

];
];
];
PCT=0;
PCT2=0;
Grupos=0;
For[r=1,r<Length[P]+1 && P[[1,r]]!=0,r=r+1,
Aux={};
Caminos=Table[0,Length[P]];
Grupos=Grupos+1;
For[m=1,m<Length[P]+1,m=m+1,
If[Capacidad[[m,r]]!=0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Min[Capacidad[[m,r]],Capacidad2[[m,r]]];
];
If[Capacidad[[m,r]]== 0 && Capacidad2[[m,r]]!=0,
Capacidad4[[m,r]]=Capacidad2[[m,r]];
];
If[Capacidad[[m,r]]!= 0 && Capacidad2[[m,r]]== 0,
Capacidad4[[m,r]]=Capacidad[[m,r]];
];
If[Capacidad4[[m,r]]!=0,
CapacidadPico[[r]]=CapacidadPico[[r]]+Capacidad4[[m,r]];
Caminos[[r]]=Caminos[[r]]+1;
Aux=AppendTo[Aux,Capacidad4[[m,r]]];
];

];
If[Caminos[[r]]>1,

CapacidadPico[[r]]=CapacidadPico[[r]]-
Max[Capacidad4[[;;,r]]];
CapacidadPico[[r]]=(CapacidadPico[[r]]/(Caminos[[r]]-1))-
Z*(StandardDeviation[Aux]/Sqrt[Length[Aux]]);
];
PCT=PCT+Max[Capacidad4[[;;,r]]];
];
"Aquí estamos prommediando la media de cada camino";

For[r=1,r<Grupos+1 ,r=r+1,
If[Caminos[[r]]>1,
PCT2=PCT2+CapacidadPico[[r]]/Grupos;
];
];

"Sería como el porcentaje en el que se redunda la capacidad del nodo destino";
If[PCT2>0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]
]]]]]]+(PCT2)*(matoriginal17[[mat[[q,1]][[1]],mat[[q,1]][[Len
gth[mat[[q,1]]]]]]]/matoriginal[[mat[[q,1]][[1]],mat[[q,1]][[Len
gth[mat[[q,1]]]]]]])/CapacidadNodo[[mat[[q,1]][[Length[mat[[
q,1]]]]]]];
```

```
];

If[PCT2== 0,
PCbase=(PCT)/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];

PC17[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=PCbase;
];

];
];

For[q=1,q<Length[mat]+1,q=q+1,

If[Length[mat[[q]]]==1&&Length[mat[[q]]]>0,
PD17[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=1;
Capacidad3={};
For[r=2,r<Length[mat[[q,1]]],r=r+1,
Capacidad3=AppendTo[Capacidad3,CapacidadNodo[[mat[[q,1]][[r]]]]];
];
For[r=1,r<Length[mat[[q,1]]],r=r+1,
For[z=1,z<Length[CapacidadesEnlaces]+1,z=z+1,
If[mat[[q,1]][[r]]<->mat[[q,1]][[r+1]]==Enlaces[[z]] || mat[[q,1]][[r+1]]<->mat[[q,1]][[r]]==Enlaces[[z]],

Capacidad3=AppendTo[Capacidad3,CapacidadesEnlaces[[z]]];
];
];
];
PC17[[mat[[q,1]][[1]],mat[[q,1]][[Length[mat[[q,1]]]]]]]=(Min[Capacidad3])/CapacidadNodo[[mat[[q,1]][[Length[mat[[q,1]]]]]]];
];
];

N[PD17//MatrixForm]
N[PC17//MatrixForm]

SumPD=Table[0,17];
SumPD1=Table[0,17];
SumPD2=Table[0,17];
SumPD3=Table[0,17];
SumPD4=Table[0,17];
SumPD5=Table[0,17];
SumPD6=Table[0,17];
SumPD7=Table[0,17];
SumPD8=Table[0,17];
SumPD9=Table[0,17];
SumPD10=Table[0,17];
SumPD11=Table[0,17];
SumPD12=Table[0,17];
SumPD13=Table[0,17];
SumPD14=Table[0,17];
SumPD15=Table[0,17];
SumPD16=Table[0,17];
SumPD17=Table[0,17];
SumPC=Table[0,17];
SumPC1=Table[0,17];
SumPC2=Table[0,17];
SumPC3=Table[0,17];
SumPC4=Table[0,17];
SumPC5=Table[0,17];
SumPC6=Table[0,17];
SumPC7=Table[0,17];
SumPC8=Table[0,17];
SumPC9=Table[0,17];
SumPC10=Table[0,17];
SumPC11=Table[0,17];
SumPC12=Table[0,17];
SumPC13=Table[0,17];
SumPC14=Table[0,17];
SumPC15=Table[0,17];
SumPC16=Table[0,17];
SumPC17=Table[0,17];

For[i=1,i<18,i=i+1,
For[j=1,j<18,j=j+1,
SumPD[[i]]=SumPD[[i]]+PD[[i,j]];
SumPD1[[i]]=SumPD1[[i]]+PD1[[i,j]];
SumPD2[[i]]=SumPD2[[i]]+PD2[[i,j]];
```

```
SumPD3[[i]]=SumPD3[[i]]+PD3[[i,j]];
SumPD4[[i]]=SumPD4[[i]]+PD4[[i,j]];
SumPD5[[i]]=SumPD5[[i]]+PD5[[i,j]];
SumPD6[[i]]=SumPD6[[i]]+PD6[[i,j]];
SumPD7[[i]]=SumPD7[[i]]+PD7[[i,j]];
SumPD8[[i]]=SumPD8[[i]]+PD8[[i,j]];
SumPD9[[i]]=SumPD9[[i]]+PD9[[i,j]];
SumPD10[[i]]=SumPD10[[i]]+PD10[[i,j]];
SumPD11[[i]]=SumPD11[[i]]+PD11[[i,j]];
SumPD12[[i]]=SumPD12[[i]]+PD12[[i,j]];
SumPD13[[i]]=SumPD13[[i]]+PD13[[i,j]];
SumPD14[[i]]=SumPD14[[i]]+PD14[[i,j]];
SumPD15[[i]]=SumPD15[[i]]+PD15[[i,j]];
SumPD16[[i]]=SumPD16[[i]]+PD16[[i,j]];
SumPD17[[i]]=SumPD17[[i]]+PD17[[i,j]];

];
];
For[i=1,i<18,i=i+1,
For[j=1,j<18,j=j+1,
SumPC[[i]]=SumPC[[i]]+PC[[j,i]];
SumPC1[[i]]=SumPC1[[i]]+PC1[[j,i]];
SumPC2[[i]]=SumPC2[[i]]+PC2[[j,i]];
SumPC3[[i]]=SumPC3[[i]]+PC3[[j,i]];
SumPC4[[i]]=SumPC4[[i]]+PC4[[j,i]];
SumPC5[[i]]=SumPC5[[i]]+PC5[[j,i]];
SumPC6[[i]]=SumPC6[[i]]+PC6[[j,i]];
SumPC7[[i]]=SumPC7[[i]]+PC7[[j,i]];
SumPC8[[i]]=SumPC8[[i]]+PC8[[j,i]];
SumPC9[[i]]=SumPC9[[i]]+PC9[[j,i]];
SumPC10[[i]]=SumPC10[[i]]+PC10[[j,i]];
SumPC11[[i]]=SumPC11[[i]]+PC11[[j,i]];
SumPC12[[i]]=SumPC12[[i]]+PC12[[j,i]];
SumPC13[[i]]=SumPC13[[i]]+PC13[[j,i]];
SumPC14[[i]]=SumPC14[[i]]+PC14[[j,i]];
SumPC15[[i]]=SumPC15[[i]]+PC15[[j,i]];
SumPC16[[i]]=SumPC16[[i]]+PC16[[j,i]];
SumPC17[[i]]=SumPC17[[i]]+PC17[[j,i]];

];
];
IM=Table[0,17,17];
IM[[;;,1]]=SumPD-SumPD1;
IM[[;;,2]]=SumPD-SumPD2;
IM[[;;,3]]=SumPD-SumPD3;
IM[[;;,4]]=SumPD-SumPD4;
IM[[;;,5]]=SumPD-SumPD5;
IM[[;;,6]]=SumPD-SumPD6;
IM[[;;,7]]=SumPD-SumPD7;
IM[[;;,8]]=SumPD-SumPD8;
IM[[;;,9]]=SumPD-SumPD9;
IM[[;;,10]]=SumPD-SumPD10;
IM[[;;,11]]=SumPD-SumPD11;
IM[[;;,12]]=SumPD-SumPD12;
IM[[;;,13]]=SumPD-SumPD13;
IM[[;;,14]]=SumPD-SumPD14;
IM[[;;,15]]=SumPD-SumPD15;
IM[[;;,16]]=SumPD-SumPD16;
IM[[;;,17]]=SumPD-SumPD17;

IM2=Table[0,17,17];
IM2[[;;,1]]=SumPC-SumPC1;
IM2[[;;,2]]=SumPC-SumPC2;
IM2[[;;,3]]=SumPC-SumPC3;
IM2[[;;,4]]=SumPC-SumPC4;
IM2[[;;,5]]=SumPC-SumPC5;
IM2[[;;,6]]=SumPC-SumPC6;
IM2[[;;,7]]=SumPC-SumPC7;
IM2[[;;,8]]=SumPC-SumPC8;
IM2[[;;,9]]=SumPC-SumPC9;
IM2[[;;,10]]=SumPC-SumPC10;
IM2[[;;,11]]=SumPC-SumPC11;
IM2[[;;,12]]=SumPC-SumPC12;
IM2[[;;,13]]=SumPC-SumPC13;
IM2[[;;,14]]=SumPC-SumPC14;
IM2[[;;,15]]=SumPC-SumPC15;
IM2[[;;,16]]=SumPC-SumPC16;
IM2[[;;,17]]=SumPC-SumPC17;

F=Table[0,17];
G=Table[0,17];

For[i=1,i<18, i=i+1,
F[[i]]=A[[i,i]];
```

```
G[[i]]=B[[i,i]];
];

For[i=1,i<18, i=i+1,
For[j=1,j<18,j=j+1,

A[[i,j]]=A[[i,j]]/F[[i]];
B[[i,j]]=B[[i,j]]/G[[i]];

A[[i,i]]=0;
B[[i,i]]=0;
];
];
Suma=Table[0,17];
Suma2=Table[0,17];
N[A//MatrixForm]
N[B//MatrixForm]

For[i=1,i<18, i=i+1,
For[j=1,j<18,j=j+1,
If[A[[i,j]]!=1,
Suma[[i]]=Suma[[i]]+A[[i,j]];
];
If[B[[i,j]]!=1,
Suma2[[i]]=Suma2[[i]]+B[[i,j]];
];
];
];

For[i=1,i<18, i=i+1,
For[j=1,j<18,j=j+1,
If[A[[i,j]]!=1,
A[[i,j]]=A[[i,j]]/Suma[[i]];
];
If[B[[i,j]]!=1,
B[[i,j]]=B[[i,j]]/Suma2[[i]];
];
];
];


N[A//MatrixForm]
N[B//MatrixForm]
```