

**Universidad  
Rey Juan Carlos**

Escuela Técnica Superior  
de Ingeniería Informática

**Grado en Matemáticas**

**Curso 2021 -2022**

**Trabajo Fin de Grado**

**GRAFOS DE COMPETITIVIDAD: DETECCIÓN DE  
COMUNIDADES Y APLICACIONES A LA  
CLASIFICACIÓN DE PAÍSES.**

Autor: Mario Canales Torres  
Tutor: Regino Criado Herrero



# Agradecimientos

Sirva este agradecimiento por toda mi etapa universitaria. Han sido 5 años que concluyen con este trabajo. Todos los mencionados habéis sido imprescindibles para haber podido completarla de la mejor manera posible.

Especial dedicatoria a toda la gente que ha estado acompañándome durante mi etapa universitaria, a mi familia y amigos, a mis compañeros y profesores.

Gracias a mis profesores del instituto, Ángel y Belén, por trasladarme esta motivación por las matemáticas.

Gracias a los profesores de la universidad por su especial dedicación. En particular a Regino, por generarme interés en la carrera desde la primera asignatura, por descubrirme el fantástico mundo de las redes complejas y por toda su paciencia e interés en este trabajo.

Gracias a mis compañeros y amigos de carrera. Todos sabemos que ha sido duro, pero habéis hecho de este camino algo muy bonito.

Gracias a mis compañeros del equipo Taekwondo Sánchez-Élez, a mi entrenador y a mis alumnos, sois luz hasta en los días nublados.

Gracias a mis padres, a mi familia y a mis amigos (que también sois familia). Por darme vuestro apoyo incondicional aún cuando menos lo merezco.

Y por último y más importante, gracias a mis abuelos, por representar todo lo que algún día quiero llegar a ser, por ser el soporte de los pasos que doy. Allá donde estéis espero que estéis orgullosos.



# Resumen

En este trabajo se revisan y aplican los conceptos teóricos correspondientes a grafos y redes complejas al estudio de la competitividad de los países según diversos indicadores utilizando la estructura de **grafo de competitividad** [1].

Son objeto de estudio los 50 países con mayor PIB (Producto Interior Bruto) en dólares en 2020, ya que son los últimos datos completos disponibles de la base de datos del Banco Mundial [2]. Se consideran los datos entre los años 2001 y 2020. A lo largo del trabajo se explican el resto de indicadores seleccionados además del PIB.

A partir de esta estructura de grafo se calcula y se compara la competitividad entre los países según varios coeficientes [3] en lo que hemos llamado *Análisis de la competitividad*. Concluimos que el índice económico que tiene mayor competitividad según todos los coeficientes es el porcentaje de crecimiento anual del PIB. No pudiendo determinar el menos competitivo por unanimidad ya que no tenemos el mismo resultado según todos los coeficientes.

Por otra parte se busca detectar comunidades mediante el algoritmo de **Girvan-Newman** [4] utilizando el grafo de competitividad según el PIB. Consideramos como separación de comunidades la separación en 5 grupos donde España se sitúa en el tercero (siendo el primer grupo formado únicamente por Estados Unidos).

Servirá este estudio para analizar desde una nueva perspectiva como han afectado los diversos sucesos económicos del siglo XXI a la competitividad económica entre países.

## Palabras clave:

- Grafos de competitividad
- Redes complejas
- Detección de comunidades
- PIB
- Girvan-Newman
- Indicadores económicos



# Índice de contenidos

<b>Índice de tablas</b>	<b>X</b>
<b>Índice de figuras</b>	<b>XIII</b>
<b>Índice de códigos</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Contexto y alcance . . . . .	1
1.2. Estructura del documento . . . . .	2
<b>2. Objetivos</b>	<b>4</b>
<b>3. Contenidos principales</b>	<b>6</b>
3.1. Grafos . . . . .	6
3.1.1. Grafos: Historia, origen y definición básica . . . . .	7
3.1.2. Conceptos básicos sobre Grafos y Redes complejas . . . . .	9
3.1.3. Rankings y grafos de competitividad . . . . .	12
3.1.4. Cuantificación de la competitividad . . . . .	15
3.1.5. Comunidades en grafos . . . . .	17
3.1.6. El algoritmo de Girvan Newman . . . . .	19
3.2. PIB y otros indicadores macroeconómicos. Definición del problema	22
3.2.1. Qué es el PIB . . . . .	22
3.2.2. Datos utilizados en el estudio . . . . .	23
<b>4. Desarrollo del trabajo y resultados</b>	<b>24</b>
4.1. Obtención y preparación de los datos . . . . .	25
4.1.1. Obtención de los datos en crudo a un fichero .xlsx (Excel)	25
4.1.2. Importación de los datos a Python y generación de los rankings . . . . .	27
4.1.3. Construcción del grafo de competitividad . . . . .	29
4.2. Análisis de competitividad . . . . .	32
4.3. Detección de comunidades . . . . .	35
<b>5. Conclusiones y trabajos futuros</b>	<b>40</b>

5.1. Conclusiones . . . . .	40
5.2. Trabajos futuros . . . . .	41
<b>6. Organización del tiempo</b>	<b>44</b>
<b>Bibliografía</b>	<b>46</b>
<b>Apéndices</b>	<b>49</b>
<b>A. Representaciones del grafo</b>	<b>51</b>
A.1. Representacion del grafo general no ponderado . . . . .	51
A.2. Representaciones de los grafos según los indicadores mencionados en 4.2 . . . . .	52
A.3. Representaciones referenciadas en 4.3 . . . . .	54
<b>B. Tabla organización del tiempo</b>	<b>56</b>
<b>C. Código utilizado</b>	<b>58</b>



# Índice de tablas

4.1. Cálculo de indicadores de competitividad en grafos con distintos indicadores . . . . .	34
---	----



# Índice de figuras

3.1. Representación del grafo . . . . .	7
3.2. Puentes de Königsberg . . . . .	8
3.3. Abstracción del problema de los puentes de Königsberg . . . . .	8
3.4. Representación del grafo dirigido . . . . .	11
3.5. Representación del grafo dirigido y ponderado . . . . .	12
3.6. Ejemplo de grafo de competitividad no ponderado . . . . .	14
3.7. Ejemplo de grafo de competitividad ponderado . . . . .	14
3.8. Ejemplo de coeficiente de clusterización.Extraído de [5] . . . . .	17
3.9. Ejemplo estructura de comunidades en redes complejas. Extraído de [6] . . . . .	18
3.10. Dendograma asociado a una partición. Extraído de [7] . . . . .	19
3.11. Cálculo de betweenness de nodos en grafo simple. Extraído de [8]	20
3.12. Cálculo de betweenness de aristas en grafo simple. Extraído de [9] .	21
4.1. Uso interfaz DataBase . . . . .	26
4.2. Mapa de los países según PIB de 2020. Obtenido en [2] . . . . .	27
4.3. Mapa del top 50 países segun su PIB en 2020. . . . .	27
4.4. Estructura del dataframe de Pandas . . . . .	28
4.5. Estructura de los rankings en Python . . . . .	29
4.6. Grafo de competitividad ponderado. . . . .	32
4.7. Grafo de competitividad ponderado referente al porcentaje de crecimiento anual de PIB. . . . .	34
4.8. Salida del algoritmo Girvan Newman en la librería networkx . . .	36
4.9. Dendograma a partir de la salida del algoritmo Girvan Newman con línea para partición de comunidades. . . . .	38
4.10. Grafo de competitividad ponderado coloreado según detección de comunidades usando algoritmo Girvan Newman. El ancho de cada arista es en función de su betweenness. . . . .	39
A.1. Grafo de competitividad no ponderado por PIB . . . . .	51
A.2. Grafo de competitividad ponderado referente al índice de exportaciones . . . . .	52

A.3. Grafo de competitividad ponderado referente al índice de importaciones . . . . .	53
A.4. Grafo de competitividad ponderado referente al GDP (PIB) per capita . . . . .	54
A.5. Grafo no ponderado con detección de comunidades . . . . .	55
B.1. Tabla de organización del tiempo para este trabajo . . . . .	57
C.1. Repositorio de github con fichero .ipnyb con el código del estudio	59



# Índice de códigos

4.1. Generación de los rankings . . . . .	29
4.2. Implementación algoritmo generación grafo de competitividad . .	31



# 1

## Introducción

Veremos en esta primera sección una introducción de los conceptos teóricos y definición del problema. A continuación, se mostrará como se estructura el presente documento.

### 1.1. Contexto y alcance

Durante décadas, los economistas se han preguntado sobre las razones que determinan el desarrollo económico de los países. Determinar las políticas económicas que impulsan el crecimiento económico es fundamental para poder promover el desarrollo de los países con menos ingresos. Sin embargo, esto no es una tarea fácil. Cada país queda caracterizado por las particularidades de su estructura económica como, por ejemplo, los sectores con más peso en la economía, las características del mercado laboral, nivel de capital humano... por tanto, nada asegura que los efectos que produzca una determinada política económica sobre un país se repliquen automáticamente en otros.

En este contexto, el producto interior bruto (PIB) se define como la medida estándar del valor agregado creado mediante la producción de bienes y servicios en un país durante un periodo determinado, habitualmente un año [10]. Así, el PIB es una variable fundamental en el estudio de la problemática anteriormente planteada, pues nos permite establecer comparaciones entre el nivel de ingreso de los países y el impacto que tienen las políticas económicas sobre el ingreso del país.

En este trabajo se toman los datos de diversos indicadores (utilizando como

principal el PIB en el año en dólares) de la base de datos del Banco Mundial [2] desde 2001 a 2020 de los 50 países que tienen un mayor PIB en el 2020. Con ello se pretende ver la evolución y competitividad de estos países durante el siglo XXI según estos indicadores y así tratar de comprender en qué indicadores existe mayor competitividad y las características comunes para su identificación en comunidades (esto último utilizando solo los datos del PIB).

Es importante mencionar que los resultados que se deriven de este trabajo en ningún caso hacen referencia de manera absoluta al bienestar material de las personas. Dado un país, las variaciones en la posición de los rankings que iremos calculando determinan variaciones en el nivel de ingreso a lo largo de los años, pero no dicen nada sobre la distribución de este u otras variables relevantes a la hora de determinar el bienestar material. Es por ello que existen otros indicadores económicos mejores para el estudio del bienestar de la población de un país como el IDH (Índice del desarrollo humano).

A la hora de elegir los diversos indicadores a estudiar hemos elegido aquellos que tienen características comunes con nuestro indicador principal (el PIB anual en dólares) para así poder hacer la comparación de competitividad con grafos de competitividad de naturalezas similares. Otra variable importante es tener la disponibilidad completa de los datos, es decir, que no falten muchos valores en nuestra consulta. Es por esto que hemos descartado varios índices, entre ellos, el IDH que mencionábamos anteriormente.

Para realizar el estudio, usaremos redes complejas basadas en grafos de competitividad [1], que explicaremos más adelante. Una vez planteados los grafos de competitividad a partir de las familias de rankings que generemos de varios indicadores, calcularemos algunas medidas clásicas de competitividad e introduciremos la posibilidad de detectar comunidades en estos grafos (utilizando únicamente el PIB), para así identificar países con PIB y tendencias de evolución de este similares.

## 1.2. Estructura del documento

Primero de todo veremos los objetivos del trabajo. Más adelante en un apartado denominado *Contenidos principales* veremos los conceptos teóricos en los que se sustenta este trabajo, así como distintos algoritmos que utilizaremos para nuestro estudio.

Después en el apartado *Desarrollo del trabajo y resultados* expondremos los resultados de aplicar los conceptos y técnicas anteriormente vistas a grafos generados con familias de rankings provenientes de distintos indicadores, separando el estudio en 2 apartados principales, el *análisis de competitividad y la detección de comunidades*.

## Capítulo 1. Introducción

Por último, veremos las Conclusiones del trabajo seguido de la organización del tiempo dedicado a hacer este trabajo.

# 2

## Objetivos

El objetivo general de este trabajo es utilizar las redes complejas y los grafos de competitividad para modelar un problema real y extraer conclusiones. A partir de estos grafos, buscaremos analizar diferentes parámetros macroeconómicos para cuantificar la competitividad y hacer una detección de comunidades.

Contamos además con los siguientes objetivos específicos:

1. Revisar conceptos básicos sobre grafos y redes complejas.
2. Aplicar las redes complejas a la modelización de un caso real.
3. Conocer los grafos de competitividad para analizar familias de rankings y aplicarlos a un problema complejo.
4. Analizar la competitividad de un modelo real de forma cuantitativa utilizando esta estructura de grafo de competitividad. Conocer parámetros para cuantificar la competitividad y utilizarlos para comparar diferentes grafos.
5. Desarrollar el código para, a partir de los datos en crudo de una consulta, conseguir familias de rankings. Desarrollar un algoritmo para generar el grafo de competitividad a partir de dichas familias. Escribir el código para generar imágenes que apoyen la presentación de los resultados obtenidos.



# 3

## Contenidos principales

En este capítulo introducimos los conceptos teóricos sobre los que se fundamenta el estudio. Separamos en 2 apartados, el primero y más extenso tratará sobre grafos y todo lo referente a esta estructura. Definiremos la estructura principal del trabajo y se explicarán los dos apartados principales en los que dividiremos la aplicación del estudio, la cuantificación de la competitividad y la detección de comunidades.

### 3.1. Grafos

En este apartado veremos un contexto histórico sobre como surge la idea de grafo y red compleja, introduciremos conceptos básicos sobre este objeto y definiremos la estructura principal de este trabajo, los **grafos de competitividad**. Después, definiremos formalmente el concepto de competitividad introduciendo unos coeficientes para cuantificarla y conoceremos lo que es una comunidad en un grafo y cómo podemos detectarlas. Extraemos algunas definiciones de uno de los libros referenciados en la asignatura cursada *Matemática discreta*. Concretamente lo referente al bloque de *Teoría de grafos*: Un semestre de matemática discreta. [11]

### 3.1.1. Grafos: Historia, origen y definición básica

En este apartado veremos una introducción básica al concepto de grafo y el contexto histórico del origen de la **teoría de grafos**.

Un grafo es una estructura que contiene una lista de vértices y una lista de aristas que conectan estos vértices. Definámolo formalmente para después ver cómo surgió esta idea.

**Definición 1.** *Un grafo simple es un par  $G = (V, E)$  formado por un conjunto finito de vértices  $V$  y un conjunto  $E$  de pares no ordenados de vértices distintos. A los elementos de  $E$  se les conoce como aristas y a los de  $V$  como vértices o nodos.*

Veamos un ejemplo simple de un grafo:

**Ejemplo 1.** Un grafo simple es por ejemplo el grafo  $G=(V,E)$  donde

$$V = \{1, 2, 3, 4\}$$

$$E = \{\{1, 2\}, \{1, 4\}, \{1, 3\}, \{2, 4\}\}$$

Y su representación es:

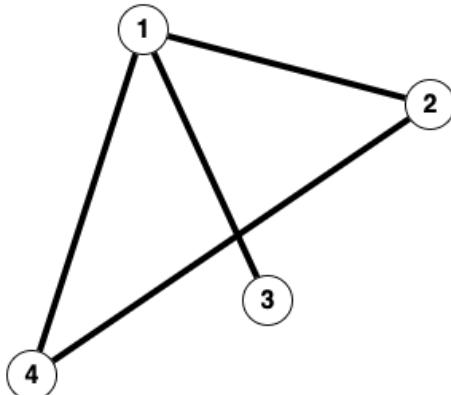


Figura 3.1: Representación del grafo

A partir de esta estructura básica, se construye la potente **teoría de grafos**. Esta teoría tiene su origen en un problema que surge en 1736 conocido como el problema de Los puentes de Konigsberg.<sup>[12]</sup> Konigsberg, actualmente llamada Kaliningrado era una antigua ciudad de Prusia atravesada por el río Pregolia. En ella se encontraban 7 puentes organizados de la siguiente manera:

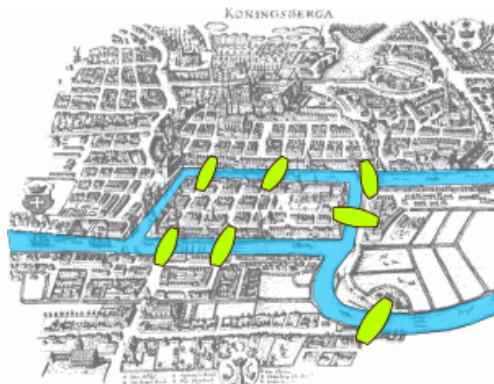


Figura 3.2: Puentes de Königsberg

El problema que se planteaban los intelectuales de la época consistía en saber si era posible visitar a pie toda la ciudad atravesando todos los puentes solo 1 vez y volver al punto de inicio.

Leonard Euler vivió en Prusia más de 20 años y trabajó en la academia Prusiana de las Ciencias [13]. En ese tiempo convivió con otros pensadores importantes de la época como Kant o Goldbach. Este cúmulo de intelectuales hicieron de Königsberg un importante enclave para la ciencia. Dada la importancia de la ciudad, el problema de los puentes de Königsberg se propagó a modo de juego entre los matemáticos e intelectuales de la época.

En 1736 Euler dio una respuesta al problema, afirmó que **no** existía posibilidad de hacer el recorrido que se proponía, para ello, utilizó una abstracción en forma de grafo de los puentes.

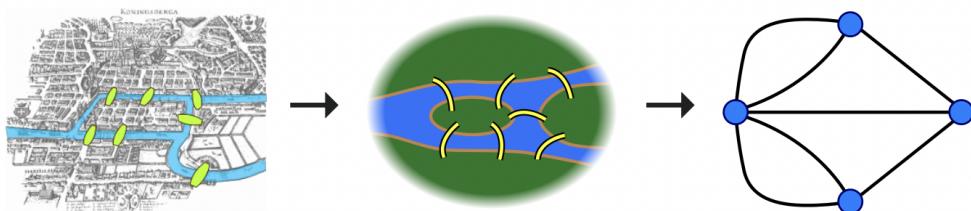


Figura 3.3: Abstracción del problema de los puentes de Königsberg

Euler enunció que un camino así solo sería posible si todos los puntos del grafo están conectados a un número par de líneas, cosa que no se cumple en este problema en ningún caso. Además, originó por primera vez el concepto de grafo y a los recorridos que cumplieran las características pedidas en el problema de los puentes de Königsberg (partir de un punto inicial y volver a ese mismo punto pasando exclusivamente solo una vez por cada arista) se les llamó **caminos Eulerianos**. Como podemos ver, esta característica del problema la podríamos considerar topológica, y en efecto, la teoría de grafos tiene fuertes relaciones con

la topología.

En este contexto, se origina la **teoría de grafos**, una herramienta con un gran potencial y que se utiliza para numerosos problemas en múltiples ámbitos.

### 3.1.2. Conceptos básicos sobre Grafos y Redes complejas

En esta sección profundizaremos en los grafos e introduciremos las **redes complejas** y los **grafos de competitividad**, conceptos clave para el desarrollo del trabajo.

La diferenciación entre los conceptos de grafo y red compleja es un poco difusa, por lo general, una **red compleja** o simplemente **red** se refiere al sistema real mientras que un **grafo** se refiere a la representación matemática de la red. En otros casos, la diferenciación entre red compleja y grafo reside únicamente en el tamaño, considerando red compleja un sistema con un tamaño grande y un grafo a un sistema con un tamaño pequeño, quedando la definición de estos tamaños al juicio de cada autor.

Históricamente esta diferenciación de los conceptos tiene sentido. En 1959 Paul Erdos y Alfréd Rényi estaban estudiando grafos [14] y propusieron un modelo en que se generaban grafos aleatorios donde las conexiones se hacían al azar. Simplemente introducían una probabilidad  $p$  de que dos nodos estuvieran conectados, es decir, cada par de nodos tenía la misma probabilidad de estar conectados. Cuando empezaron a utilizarse los grafos para modelar problemas reales los científicos se dieron cuenta de que los resultados obtenidos eran muy diferentes a lo que se obtenían estudiando grafos aleatorios [15] y fue cuando se les denominó como redes complejas. Las propiedades de estas redes complejas son distintas a los grafos aleatorios porque van muy ligadas al sistema que modelan.

En los estudios que se han hecho desde entonces, se ha comprobado que al representar en una red compleja las características de un sistema, a partir de un estudio eficiente y exhaustivo de esta estructura se puede obtener información muy valiosa.

En lo que sigue, estaremos manejando estos dos conceptos indistintamente. Continuamos introduciendo conceptos importantes sobre grafos.

**Definición 2.** El **grado** de un nodo o vértice en un grafo  $G$  es el número de aristas incidentes a dicho vértice, es decir, el número de vértices a los que está conectado. Denotaremos el grado de un vértice  $x$  como  $\deg(x)$  por su nombre en inglés "degree".

Por ejemplo, en el grafo de ejemplo 3.3 que modela el problema de los puentes de Konigsberg, el grado del vértice 1 es  $\deg(1) = 3$  pues está conectado con los

vértices 2, 3 y 4. Mientras que el grado del vértice 3 es  $\deg(3) = 1$  ya que sólo está conectado con el vértice 1.

Esta definición nos servirá más adelante para calcular diferentes coeficientes de competitividad. Continuemos con otras definiciones básicas.

**Definición 3.** La **matriz de adyacencia** de un grafo  $G = (V, E)$  es una manera única de determinar la estructura de un grafo. La definición de esta matriz es sencilla. La matriz  $A(G)$  de adyacencia de un grafo  $G = (V, E)$  tiene los elementos de cada posición  $i, j$  definidos como:

$$a_{ij} = \begin{cases} 1 & \text{si } \{i, j\} \in E \\ 0 & \text{si } \{i, j\} \notin E \end{cases}$$

La diagonal de esta matriz estará compuesta por ceros (en algunos grafos se consideran aristas con principio y final en el mismo nodo, no es el caso en este trabajo). En los grafos que estamos contemplando la matriz será **simétrica**, ya que las aristas no tienen sentido. Añadimos pues, el concepto de grafo dirigido, donde las aristas **sí** tienen sentido.

**Definición 4.** Un **grafo dirigido** es un grafo en el cual las aristas tienen dirección. En este caso el conjunto de aristas  $E$  se presenta como pares ordenados  $(u, v)$ . Veamos un ejemplo.

**Ejemplo 2.** Un grafo dirigido es por ejemplo el grafo  $G = (V, E)$  donde

$$V = \{1, 2, 3, 4\}$$

$$E = \{(2, 1), (1, 4), (3, 1), (2, 3)\}$$

Y su representación es [3.4](#).

Además de esta definición, es de suma importancia definir el **grafo ponderado**.

**Definición 5.** Un **grafo ponderado** es un grafo (dirigido o no) en el que las aristas llevan asociado un peso. Este peso en grafos que modelan problemas reales usualmente pueden significar el coste de realizar ese camino o la distancia que conecta los 2 nodos.

Formalmente lo podemos definir como 3 conjuntos,  $G = (V, E, W)$  donde  $V$  y  $E$  son igual que en las definiciones anteriores y  $W$  es el peso de cada arista, en el mismo orden en que aparece en  $E$ . La cardinalidad del conjunto  $E$  es **necesariamente igual** que la del conjunto  $W$ . Si queremos representar el grafo mediante

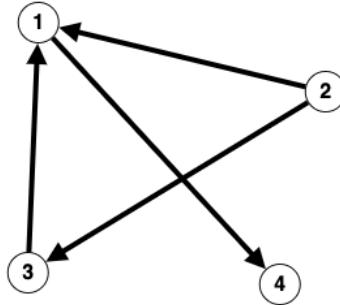


Figura 3.4: Representación del grafo dirigido

una matriz de incidencia, la podemos hacer sustituyendo los unos y ceros por los pesos correspondientes. Veamos una definición más formal.

Sea  $G=(V,E,W)$

$$V = \{v_1, \dots, v_n\}, E = \{e_1, \dots, e_m\}, W = \{w_1, \dots, w_m\},$$

entonces la matriz de adyacencia  $A$  del grafo ponderado se define como

$$a_{ij} = \begin{cases} w_k & \text{si } \{i, j\} \in E \text{ y } \{i, j\} = e_k \\ 0 & \text{si } \{i, j\} \notin E \end{cases}$$

Veamos por último, un ejemplo de un grafo dirigido y ponderado, viendo además su matriz de adyacencia:

**Ejemplo 3.** El grafo dirigido y ponderado  $G=(V,E,W)$

$$V = \{1, 2, 3, 4\}$$

$$E = \{\{4, 1\}, \{2, 3\}, \{1, 2\}, \{1, 3\}\}$$

$$W = \{2, 1, 1, 5\}$$

tiene como matriz de adyacencia:

$$A = \begin{pmatrix} 0 & 1 & 5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} \quad (3.1)$$

Y su representación es la que aparece en la figura 3.5

Una vez conocidos estos conceptos, podemos dar paso al objeto matemático objeto de este trabajo: los grafos de competitividad.

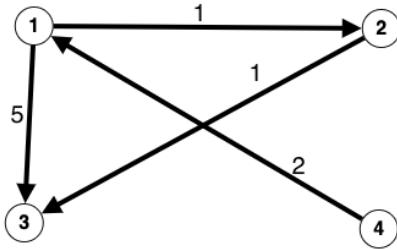


Figura 3.5: Representación del grafo dirigido y ponderado

### 3.1.3. Rankings y grafos de competitividad

El concepto de competitividad es un concepto difuso, en la RAE podemos encontrar la definición de competitividad como «capacidad para competir». Claramente esta definición no nos aporta un significado conciso ni cuantificable.

En el entorno deportivo la competitividad de una competición es un concepto importante. A mayor competitividad podemos encontrar más incertidumbre para estimar el resultado final. Una alta competitividad significa que hay una alta incertidumbre para poder predecir qué participante de la competición acabará en cada posición. Además de en el entorno deportivo, la palabra competitividad se usa en muchos más dominios, como la competitividad por los recursos en biología, la competitividad entre empresas por obtener el mayor número de clientes, o, en el caso que nos compete, la competitividad por distintos países por su PIB u otros indicadores económicos.

Existen estudios donde se analiza la competitividad en competiciones deportivas (como por ejemplo en fútbol [3]) pero en este caso vamos a estudiarla en el contexto económico del PIB.

Como hemos comentado, en este apartado veremos el objeto principal de este trabajo, el **grafo de competitividad** (extraemos varias definiciones del documento [1]). Para ello necesitamos utilizar los conceptos de grafos previamente explicados y definir un concepto nuevo, el concepto de **ranking**, que será la base de las estructuras que definiremos en este apartado.

**Definición 6.** Dado un conjunto de elementos finitos  $\mathcal{N} = \{1, \dots, n\}$  con  $n \in \mathbb{N}$  que llamamos nodos definimos un **ranking**  $c$  de  $\mathcal{N}$  como una biyección  $c : \mathcal{N} \rightarrow \mathcal{N}$ . Así pues podemos identificar un ranking como un vector de  $\mathcal{N}^n = \mathcal{N} \times \mathcal{N} \times \dots \times \mathcal{N}$ .

De esta forma si tomamos un ranking  $c : \mathcal{N} \rightarrow \mathcal{N}$  podemos identificarlo como vector siendo  $c \equiv (i_1, i_2, \dots, i_n) \in \mathcal{N}^n$  donde  $c(1) = i_1$ ,  $c(2) = i_2 \dots c(n) = i_n$ .

Tomando el ranking  $c$ , decimos que  $i \prec_n j$  si el nodo  $i$  aparece antes que el

nodo  $j$  en el vector  $c$ , es decir  $c(i) < c(j)$ .

Esta definición de ranking es la formalización de la definición que todos podemos conocer en diferentes competiciones deportivas. Por lo general se suelen construir los rankings en torno a algún parámetro a partir del cual podamos ordenar, por ejemplo, en una competición de fútbol los puntos (siendo el primer elemento del ranking el equipo con más puntos). En nuestro caso el parámetro que utilizaremos para construir los rankings sera principalmente el PIB por año en cada país (utilizaremos para otros rankings otros indicadores), aunque esto lo detallaremos más adelante.

Dada la definición de ranking, necesitamos definir qué es **competir** para así llegar al objeto principal de este trabajo, los **grafos de competitividad**.

**Definición 7.** Dado un conjunto finito de rankings  $\mathcal{R} = \{c_1, c_2, \dots, c_r\}$  decimos que los nodos  $(i, j) \in \mathcal{N}$  **compiten** si existe  $c_s, c_t \in \{1, 2, \dots, r\}$  tal que  $i \prec_{c_s} j$  pero  $j \prec_{c_t} i$ . Esto quiere decir que estos nodos intercambian sus posiciones relativas (en este caso, el nodo  $i$  «adelanta» al nodo  $j$ ).

Teniendo ya la definición de ranking y de qué significa competir, tomando una familia de rankings la podemos **asociar a un grafo** para obtener información sobre como se estructura la competitividad entre nodos.

**Definición 8.** Sea  $\mathcal{R} = \{c_1, c_2, \dots, c_r\}$  una familia de rankings de los nodos  $\mathcal{N} = \{1, \dots, n\}$ . Definimos el **grafo de competitividad** de la familia de rankings  $\mathcal{R}$  como el grafo no dirigido  $G_c(\mathcal{R}) = (\mathcal{N}, E_R)$  donde el conjunto de aristas  $E_R$  es el conjunto dado por la siguiente regla: hay una arista entre los nodos  $i$  y  $j$  si compiten.

Podemos **extender esta definición** de una forma muy natural para cuantificar el número de veces que compiten los nodos.

**Definición 9.** En las mismas condiciones que la definición anterior, definimos el **grafo de competitividad ponderado** como el grafo no dirigido  $G_c(\mathcal{R}) = (\mathcal{N}, E_R)$ , siendo  $R$  y  $\mathcal{N}$  igual que antes y considerando que la regla para construir el conjunto de aristas  $E_R$  es la siguiente: se añade una unidad a la ponderación de la arista entre los nodos  $i$  y  $j$  cada vez que compiten, es decir, se añade 1 a las posiciones de la matriz  $\{i,j\}$  y  $\{j,i\}$ .

Con estas definiciones veamos un ejemplo:

**Ejemplo 4.** Consideramos  $\mathcal{N} = \{1, \dots, 5\}$  y los siguientes rankings de  $\mathcal{N}$

$$c_1 \equiv (1, 2, 3, 4, 5)$$

$$c_2 \equiv (1, 3, 5, 2, 4)$$

$$c_3 \equiv (1, 5, 2, 4, 3)$$

$$c_4 \equiv (1, 2, 4, 3, 5)$$

El grafo de competitividad no ponderado asociado a esta familia de rankings es el siguiente:

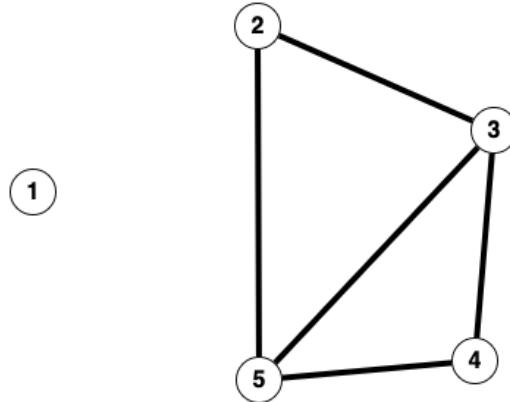


Figura 3.6: Ejemplo de grafo de competitividad no ponderado

Mientras que, si tenemos en cuenta la ponderación de las aristas, el grafo de competitividad ponderado referente a la familia de rankings es:

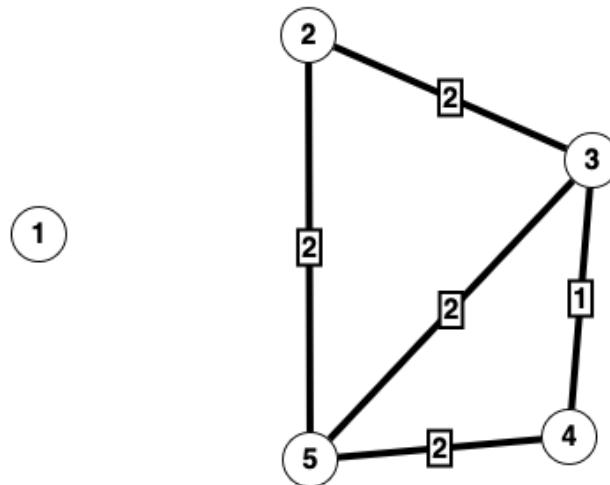


Figura 3.7: Ejemplo de grafo de competitividad ponderado

Veamos un par de anotaciones importantes en estas definiciones:

- Para calcular las aristas hay que ir viendo en cada ranking que elementos han intercambiado la posición respecto al ranking anterior. Por ejemplo, en [3.6](#) la arista del vértice 3 al 4 se obtiene tras comprobar que en el ranking  $c_2$  tenemos que  $3 \prec_{c_2} 4$  y en el ranking  $c_3$  ocurre lo contrario, ya que  $4 \prec_{c_3} 3$  es decir, intercambian sus posiciones relativas.
- El vértice 1 no está conectado, esto se debe a que no intercambia la posición con ningún otro nodo, es decir, no compite en ningún ranking con el resto de nodos. Hemos elegido esta configuración como ejemplo porque es representativo con el grafo de competitividad principal (por PIB) que veremos en este estudio, donde existe un país (USA) que no compite en ningún ranking de la familia. [4.6](#)
- Como es obvio, la forma o «topología» del grafo no cambia del ponderado al no ponderado, simplemente se añaden los números correspondientes a la ponderación en cada arista.

Para ver un algoritmo para la construcción del grafo de competitividad, consultar el apartado [4.1.3](#) donde se detalla el que hemos desarrollado y utilizado en este trabajo.

### 3.1.4. Cuantificación de la competitividad

Como hemos comentado en [3.1.3](#) el concepto de competitividad es bastante difuso y está presente en muchos sistemas reales, es por ello que se han propuesto medidas que cuantifiquen esta competitividad.

En lo que sigue en este apartado, enunciaremos algunas medidas de competitividad que nos sirven para el problema que nos atañe, cuantificar la competitividad dada una familia de rankings  $\mathcal{R}$ .

Las definiciones son extraídas de [\[3\]](#).

Sea  $\mathcal{R} = \{c_1, c_2, \dots, c_r\}$  una familia de rankings, sea el conjunto de nodos  $\mathcal{N} = \{1, \dots, n\}$  y el grafo de competitividad  $G_c(\mathcal{R}) = GC(R)$  (para simplificar notación) el grafo de competitividad, se definen los siguientes índices:

- **Normalized mean degree (NMD):** En español, grado medio normalizado. Definimos el NMD de una familia de rankings  $\mathcal{R}$  como la suma de los grados de todos los nodos en el grafo de competitividad  $GC(\mathcal{R})$  dividido por la suma de los grados más altos posibles de todos los nodos (n nodos conectados a n-1 nodos como máximo)

$$ND(\mathcal{R}) = \frac{1}{n(n+1)} \sum_{i \in \mathcal{N}} deg(i). \quad (3.2)$$

Este parámetro nos indica en qué proporción encontramos en el grafo cada posible relación entre nodos, es decir, aplicado a grafos de competitividad el NMD nos aporta una idea global sobre el número de veces en que dos posibles competidores han intercambiado sus posiciones relativas en la correspondiente familia de rankings.

Por ejemplo en el grafo 3.6 tenemos que su NMD es de  $\frac{1}{2}$  ya que la suma de los grados es 10 y la máxima posible es 20 ( $n = 5$ ).

Decimos que  $\mathcal{R}$  es más competitivo que  $\mathcal{S}$  respecto al NMD si  $ND(\mathcal{R}) > ND(\mathcal{S})$

- **Normalized mean strength (NMS):** En español, fuerza media normalizada. Este coeficiente se aplica sobre el **grafo de competitividad ponderado**. La fuerza (strength) de un nodo en un grafo ponderado es la suma de los pesos de sus aristas incidentes. Definimos el coeficiente NMS de la familia de rankings  $\mathcal{R}$  como la suma de todos los pesos de sus aristas en el grafo de competitividad  $GC(\mathcal{R})$  dividido entre la suma de todos los posibles bordes con su máximo peso posible:

$$NS(\mathcal{R}) = \frac{w(E_{\mathcal{R}}^e)}{\binom{n}{2}(r-1)}. \quad (3.3)$$

Donde  $w(E_{\mathcal{R}}^e)$  denota la suma de todos los pesos de las aristas del grafo de competitividad. El denominador:  $\binom{n}{2}(r-1)$  denota el máximo peso posible, ya que  $r-1$  es el número de rankings y  $\binom{n}{2}$  expresa las posibilidades en las que se pueden combinar los nodos (en este caso sería el número máximo de nodos que compiten dado un ranking).

En el grafo 3.7 tenemos una NMS de  $\frac{3}{10}$ , ya que la suma de los pesos es de 9 y tenemos  $r=4$  y  $n=5$  y por tanto  $\binom{n}{2}(r-1) = \binom{5}{2}(3) = 30$ .

- **Coeficiente de clusterización (CC):** Se define una clique como un conjunto de nodos conectados mutuamente entre ellos. Este coeficiente mide cuantos nodos del grafo tienden a agruparse juntos, a formar una clique. El coeficiente de clusterización de un nodo  $i$ ,  $C_i$  se define como:

$$C_i = \frac{e_i}{\binom{k_i}{2}}. \quad (3.4)$$

Donde  $k_i$  es el número de vecinos del nodo  $i$  y  $e_i$  es el numero de pares conectados entre los vecinos de  $i$ . El término  $\binom{k_i}{2}$  representa todos los posibles pares entre los vecinos de  $i$ .

En 3.8 se calcula el coeficiente de clusterización del nodo verde. Como vemos el nodo verde tiene 3 vecinos. Para calcular el coeficiente de clusterización tenemos que ver las conexiones entre esos vecinos. En el primer caso están todos conectados, es por ello que el coeficiente es 1. En el segundo caso

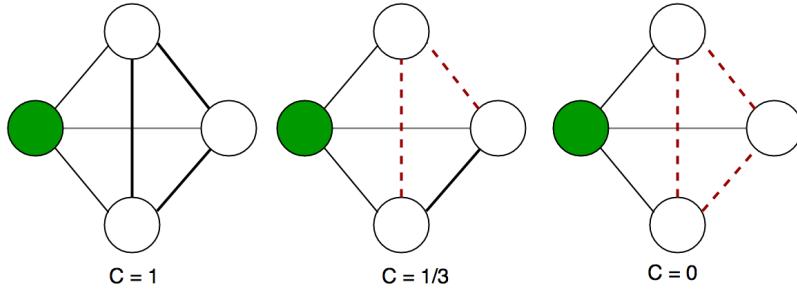


Figura 3.8: Ejemplo de coeficiente de clusterización.Extraído de [5]

(nótese que las líneas rojas son las aristas inexistentes) solo hay un par de vecinos conectados de 3 posibles, por ello el coeficiente es  $\frac{1}{3}$ . Por último, el último caso no tiene vecinos del nodo verde conectados y por tanto el coeficiente es 0.

Dada una familia de rankings  $\mathcal{R}$ , el coeficiente de clusterización de  $\mathcal{R}$  es la media de todos los coeficientes de clusterización de los nodos del grafo de competitividad asociado a esa familia de rankings,  $G_c(\mathcal{R})$ . Es decir:

$$C(\mathcal{R}) = \frac{1}{n} \sum_{i \in N} C_i. \quad (3.5)$$

- Además, existen otros coeficientes como el Coeficiente de correlación generalizada  $\tau$  de Kendall, el número maximo normalizado de componentes conectadas, el tamaño normalizado máximo de las cliques... que no utilizaremos en este caso.

### 3.1.5. Comunidades en grafos

En una red compleja o grafo, una **comunidad** se puede definir como un conjunto de nodos que están mas densamente conectados entre ellos que con el resto de la red.

Cuando estamos hablando de una red compleja que modela un sistema real, esta estructura de comunidades es de gran importancia ya que se espera que los nodos que están contenidos dentro de una misma comunidad compartan atributos o características comunes entre ellos. No existe una definición exacta de que es una comunidad y por ello hay varias formas de separar un grafo en distintas comunidades.

Este problema se conoce como **clustering** o **detección de comunidades** y no solo se utiliza para grafos, sino que también se utiliza para agrupar o separar datos de distintas naturalezas en problemas estadísticos, de «Data Science» o de

análisis de datos. Además, es uno de los problemas más estudiados en los términos de la inteligencia artificial, dando lugar a algoritmos muy importantes como el tan conocido algoritmo de las k-medias. [16]

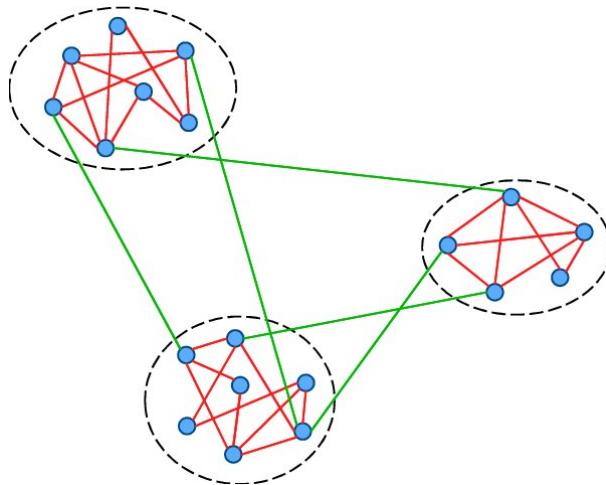


Figura 3.9: Ejemplo estructura de comunidades en redes complejas. Extraido de [6]

En la figura 3.9 vemos un ejemplo de una red compleja dividida en 3 comunidades. Podemos ver que la mayoría de las conexiones son **intra-comunidad**, esto es lo que determina que los nodos pertenecientes a esta comunidad comparten más con los nodos de su misma comunidad que con el resto, independientemente de cual sea el criterio para establecer una arista entre 2 nodos. Podemos ver que esta división de comunidades es **sin solapamiento**, es decir, no existen nodos pertenecientes a más de una comunidad.

Podemos definir una **partición** como una división de una red en comunidades o «clusters». La manera de hacer una partición a un grafo es objeto de estudio y se han desarrollado diversos algoritmos. Las divisiones dada una partición pueden estar a su vez jerarquizadas y poder unirse para formar comunidades más grandes. Estas estructuras se pueden representar mediante un esquema llamado **dendograma**.

En la figura 3.10 vemos como encontramos 3 comunidades en una red compleja. Mediante la estructura jerárquica mostrada en el dendograma, podríamos hacer divisiones más pequeñas, realizando la división a partir de un nivel más profundo.

Para hacer la división por comunidades existen varios algoritmos, y como la definición de comunidad no está clara, las soluciones pueden divergir. Podemos encontrar el algoritmo de cortes mínimos [17], algoritmos de maximización de modularidad [18] o el que vamos a tratar en este trabajo, el algoritmo de **Girvan-Newman**.[4].

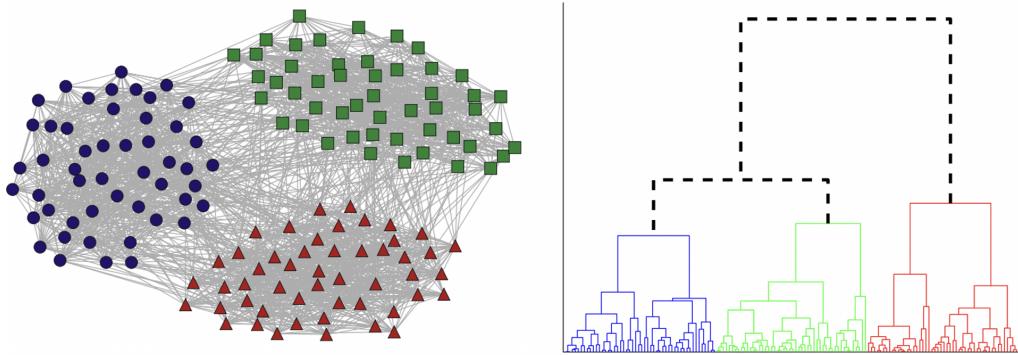


Figura 3.10: Dendograma asociado a una partición. Extraído de [7]

Los algoritmos utilizan diversas medidas que determinan qué características se deben resaltar para hacer las comunidades. Entre otras medidas, existen algoritmos que se fijan en la distancia entre nodos, en la densidad (número de aristas presentes) o en la centralidad por intermediación (conocida como **betweenness**).

### 3.1.6. El algoritmo de Girvan Newman

El algoritmo de Girvan Newman [4] es uno de los métodos más usados para detectar comunidades en redes complejas. Es un algoritmo **jerárquico**, es decir, tiene como objetivo establecer una jerarquía entre los distintos agrupamientos con diferentes niveles y se puede visualizar mediante un dendrograma 3.10.

La medida que utiliza este algoritmo es la centralidad por intermediación, conocida como **betweenness**. Para definir la betweenness antes debemos definir otro término.

**Definición 10.** En teoría de grafos, el **camino más corto** consiste en encontrar el camino (recorrido de vértices unidos por aristas con principio y final en nodos) entre dos vértices tal que la suma de los pesos (si no es ponderado cada arista se considera de peso 1) sea mínima. Este problema **no tiene necesariamente una única solución**.

Veremos un algoritmo que calcula el camino más corto en el apartado de Desarrollo y Resultados 4, el algoritmo de Dijkstra. Dada la definición de camino más corto, podemos definir la betweenness. Diferenciamos entre la betweenness de nodo y de arista. Primeramente definimos la de nodo para después extender esta definición a la betweenness de arista, que es la que realmente nos interesa.

**Definición 11.** La **betweenness** o centralidad por intermediación de un **vértice** es una medida de centralidad que cuantifica el número de veces que un nodo se

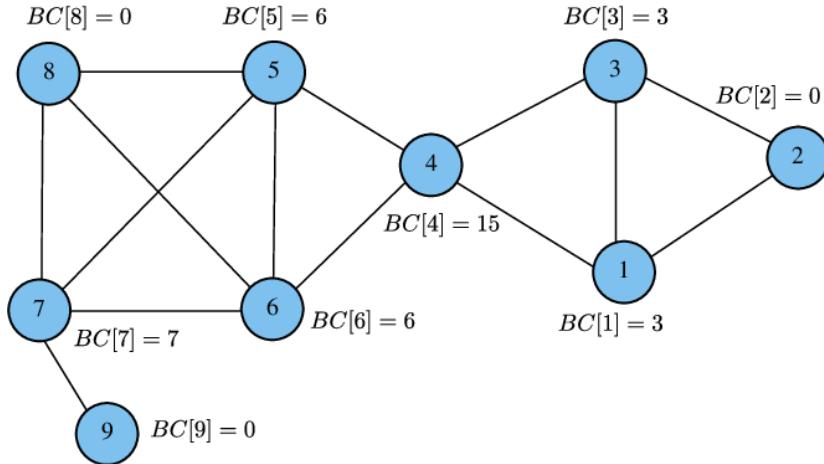


Figura 3.11: Cálculo de betweenness de nodos en grafo simple. Extraído de [8]

encuentra en el camino más corto entre otros dos nodos. Formalmente, sea el grafo  $G=(V,E)$  un grafo, la betweenness  $B_C(i)$  de un nodo  $i$  se define como:

$$B_C(i) = \sum_{j \neq k \in V} \frac{b_{jik}}{b_{jk}}. \quad (3.6)$$

Donde  $b_{jk}$  es el número de caminos más cortos desde el nodo  $j$  hasta el nodo  $k$  y  $b_{jik}$  es el número de caminos más cortos desde  $j$  hasta  $k$  que pasan a través del nodo  $i$ . Esta ecuación mide el número de caminos más cortos en los que aparece el nodo  $i$  entre todos los posibles.

En 3.11 Podemos ver un ejemplo de un grafo sencillo donde se calcula la betweenness de los vértices. Como podemos comprobar, el nodo 4 es el que haría de intermediario para conectar dos hipotéticas comunidades, la de los nodos que quedan a la izquierda de él y la de los nodos que quedan a la derecha de él. En [8] podemos encontrar varios algoritmos optimizados para poder calcular la betweenness de los vértices de un grafo con la menor complejidad posible.

En el algoritmo de Girvan-Newman se extiende esta definición para el caso de las aristas. Se define la **betweenness de aristas** de una arista como el numero de caminos más cortos que pasan por esa arista.

Podemos ver en 3.12 que tenemos una partición en comunidades clara. Para calcular la betweenness de la arista que separa las dos comunidades tenemos que calcular el número de caminos más cortos en los que interviene. Es claramente 25, ya que interviene únicamente en todos los caminos entre nodos de distintas comunidades, es decir, el número de caminos sería el producto del número de nodos de ambas comunidades, en este caso  $5 \times 5 = 25$ .

En adelante cuando hablemos de betweenness nos referiremos directamente a

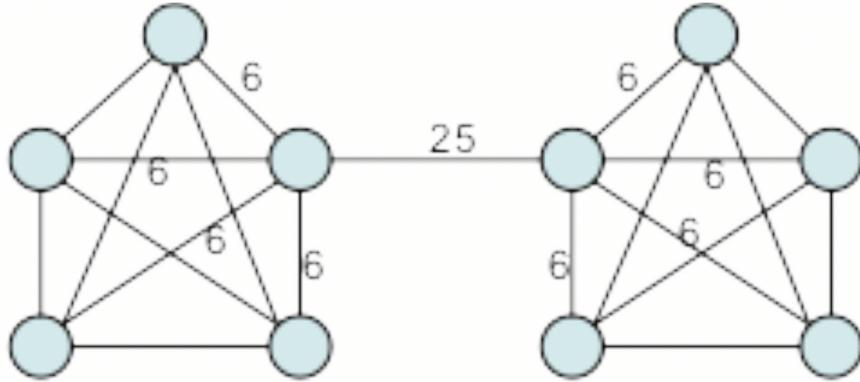


Figura 3.12: Cálculo de betweenness de aristas en grafo simple. Extraído de [9]

betweenness de aristas según lo anteriormente definido.

La idea del algoritmo de **Girvan Newman** es que si las comunidades tienen pocas conexiones extracomunitarias, es decir, que conectan los nodos de una comunidad con los nodos de otra comunidad diferente, entonces los caminos más cortos entre las diferentes comunidades deben pasar por esas pocas aristas. Por tanto, los enlaces que conectan comunidades tendrán una betweenness alta. Mediante la eliminación de estos enlaces vamos separando grupos y vamos obteniendo una estructura de comunidad. En 3.12 la primera arista que eliminaríamos sería la etiquetada como 25, que nos daría de resultado las dos comunidades que identificamos gráficamente de forma directa.

Es notable observar que según la idea anterior, con el algoritmo de Girvan Newman iremos obteniendo una estructura jerárquica y deberemos elegir un nivel por el que hacer la división en grupos para obtener la estructura de comunidades.

Los pasos del algoritmo de detección de comunidades de Girvan Newman son los siguientes:

1. Se calcula la betweenness de todas las aristas de la red
2. Eliminamos la arista con mayor betweenness
3. Se recalcula la betweenness de todas las aristas
4. Los pasos 2 y 3 son repetidos hasta que no queden más enlaces

Existen varias formas de implementar este algoritmo, una de ellas es usar recursividad. Se aplica el algoritmo en todas las comunidades que vamos obteniendo cada vez que eliminamos una arista. Esta versión no tiene por qué ser la más óptima.

En la librería de python networkx existe una implementación de este algoritmo [19] que será la que se utilice en este trabajo. También existe un algoritmo para calcular únicamente la betweenness que utilizaremos para utilizar en algún gráfico.

## 3.2. PIB y otros indicadores macroeconómicos. Definición del problema

En este apartado explicaremos cómo vamos a abordar el problema, de dónde sacamos los datos y qué pasos vamos a dar. En concreto definiremos y explicaremos por qué el indicador principal elegido es el PIB, y la fuente de los datos. Más adelante propondremos la idea principal del problema, que consiste en generar el grafo de competitividad y realizar el estudio de competitividad y comunidades.

### 3.2.1. Qué es el PIB

El producto interior bruto (PIB) se define como la medida estándar del valor agregado creado mediante la producción de bienes y servicios en un país durante un periodo determinado [10]. En este caso, utilizamos el PIB en dólares por año así que podríamos interpretar el PIB como la cuantificación en dólares americanos (US\$) del conjunto de bienes y servicios producidos en un país durante un año.

Históricamente este índice ha sido el más utilizado para determinar el nivel de riqueza de un país. Este índice ha permitido a los economistas saber si la economía de un país esta en crecimiento, estancada... y en lo que a este trabajo compete, poder compararlo con economías de otros países utilizando una cuantificación global, en este caso, los dólares. Es por esto por lo que el PIB será el indicador principal en lo que sigue.

Para la obtención de datos utilizaremos los datos públicos de la base de datos del banco mundial (World Development Indicators). [2]

Concretamente estamos viendo el indicador GDP (current US\$). GDP (gross domestic product) son las siglas en inglés para PIB. En este caso disponemos del dato en dólares.

Hemos elegido como índice principal el PIB porque, además de ser un dato accesible fácilmente cuantifica muy bien los factores macroeconómicos de las economías de los diferentes países. Consideramos que es explicativo sobre el valor monetario de la producción de los bienes y servicios de cada país.

Hay algunos datos que no se incluyen en el cálculo del PIB, simplemente porque no se pueden contabilizar ni conocer. Por ejemplo, los bienes de autoconsumo

o la llamada economía sumergida.[20]

### 3.2.2. Datos utilizados en el estudio

Como hemos visto antes, el indicador principal elegido es el PIB anual. No obstante, aún debemos indicar cuáles son los países que serán objeto de estudio y que otros índices utilizaremos para el apartado en que se cuantifica y compara la competitividad.

Los países que serán los nodos de todos los grafos de competitividad y de las familias de rankings que se usarán en este trabajo serán los 50 países con mayor PIB anual desde el 2001 al 2020 (año correspondiente al último dato disponible del banco mundial). En otras fuentes podíamos encontrar también el dato de 2021 pero estos datos no estaban disponibles para todos los países y estaban basados en estimaciones que no hemos considerado igual de fiables que las del banco mundial.

Es notable saber que hay países que no están presentes en estos rankings, como es el caso de Venezuela, que desde el 2014 y debido a políticas internas del país, se dejaron de enviar datos. Esto se debió a la decisión de salir del Banco Mundial y FMI. [21]

# 4

## Desarrollo del trabajo y resultados

Como hemos detallado anteriormente, el objetivo de este trabajo consiste en analizar la competitividad de los países de acuerdo a algunos indicadores como su PIB anual y estudiar cómo se han desarrollado durante el siglo XXI.

Para ello utilizaremos la estructura de los grafos de competitividad para obtener información importante sobre el modelo macroeconómico que estamos modelando.

En este apartado explicaremos detalladamente las fases de desarrollo y los resultados del estudio.

En una primera fase tendremos que recolectar, limpiar y modificar los datos para poder analizarlos correctamente y aplicar un estudio de competitividad. Este apartado comienza en la obtención y limpieza de los datos en crudo. Una vez hecho esto tendremos que hacer operaciones con estos datos para obtener los 20 rankings (uno por año) para así poder generar los grafos de competitividad.

Una vez tengamos los grafos de competitividad computaremos y compararemos la competitividad según varios coeficientes.

Por último, con nuestro grafo de competitividad correspondiente al PIB anual, ejecutaremos el algoritmo de Girvan-Newman para encontrar la estructura de comunidades subyacente al grafo generado.

## 4.1. Obtención y preparación de los datos

En este apartado veremos el proceso realizado desde la obtención de los datos en crudo tras la consulta a la base de datos hasta la generación del grafo de competitividad relativo a esos datos.

### 4.1.1. Obtención de los datos en crudo a un fichero .xlsx (Excel)

Como hemos comentado antes, para este trabajo utilizaremos los bancos obtenidos de la base de datos de indicadores del Banco Mundial (World Development Indicators) [2].

La base de datos (World Databank) es accesible mediante una interfaz gráfica sencilla donde podemos seleccionar el dato que queremos analizar ya que hay disponibles bastantes indicadores económicos para consultar.

Seleccionamos el indicador GDP (current US\$) para todos los países (apartado «Countries»). Es importante destacar que la World DataBank también ofrece la posibilidad de consultar lo que ellos llaman «Aggregates» que son distintas comunidades que no son países (África Este, Europa Central, Mundo Árabe...). Seleccionamos estrictamente los países. Para ello simplemente eliminamos todas las selecciones de la pestaña «aggregates» y seleccionamos todas de «countries».

Por otra parte, en el apartado «Time» seleccionamos los años desde 2001 a 2020, pues los últimos datos disponibles, 2021, no están completos. Podemos ver un ejemplo de uso de la interfaz en [4.1](#).

Una vez hecha la query (consulta) tenemos la opción descargar los datos en varios formatos como Excel ( .xlsx), CSV... Nosotros hemos elegido el formato .xlsx, pues desde el programa Excel haremos un preprocesamiento sencillo de los datos.

Con los datos descargados, en el programa excel ordenamos los datos según el dato (columna) del PIB del 2020 (vamos a usar el top 50 países del 2020). Notemos que si ordenamos en cualquier otro año la lista de países puede ser diferente, ya que los elementos de las últimas posiciones del ranking pueden competir con elementos que estén fuera del ranking elegido, es decir, intercambiar sus posiciones. Una vez hecho esto eliminamos todas las filas a partir de la fila número 50 y así tendremos el histórico de datos de los 50 países que nos interesan.

Lo que hemos hecho en este paso es eliminar la lista de países que no vamos a estudiar. Podemos comparar dos mapas, el que muestra la web del World Bank donde se incluyen todos los países y uno que elaboramos con los datos que vamos a utilizar en nuestro trabajo. Ambos gráficos atienden al año 2020.

## 4.1. Obtención y preparación de los datos

The screenshot shows the DataBank interface for the World Development Indicators. At the top, there are tabs for Variables, Layout, Styles, Save, Share, and Embed. Below these are sections for Database (Available 85, Selected 1) and Country (Available 266, Selected 217). A search bar allows entering keywords. A list of countries is displayed, with checkboxes next to each name. Some countries have their names followed by descriptive text and small icons. A sidebar on the right lists categories like Africa Eastern and Southern, Albania, Algeria, Andorra, Antigua and Barbuda, Argentina, Aruba, Austria, and so on. At the bottom, there's a 'Create Custom Group' button and another section for Series (Available 1445, Selected 1).

Figura 4.1: Uso interfaz DataBase

Ambos mapas muestran información semejante, en el segundo ([4.3](#)) solo tenemos los datos (regiones coloreadas) de los 50 países con mayor PIB en 2020, que son exactamente los que vamos a utilizar, mientras que en el primero ([4.2](#)) tenemos los datos de todos los países del WorldBank.

Es importante tener claro la idea de que para todos los indicadores vamos a utilizar los mismos países. Es decir, si utilizáramos el indicador «GDP per cápita» con los 50 países con mayor valor en 2020, la lista de países sería diferente, ya que por ejemplo encontraríamos el país Luxemburgo, que no aparece en el ranking de los 50 países con mayor PIB en 2020. Para ello, en las consultas que hagamos para el resto de rankings, en el apartado «Countries» seleccionaremos únicamente los 50 países que hemos mencionado.

Teniendo los datos preparados, es momento de utilizarlos desde código para comenzar a manipularlos y así generar las familias de rankings que sean la base de la estructura final que buscamos: el grafo de competitividad.

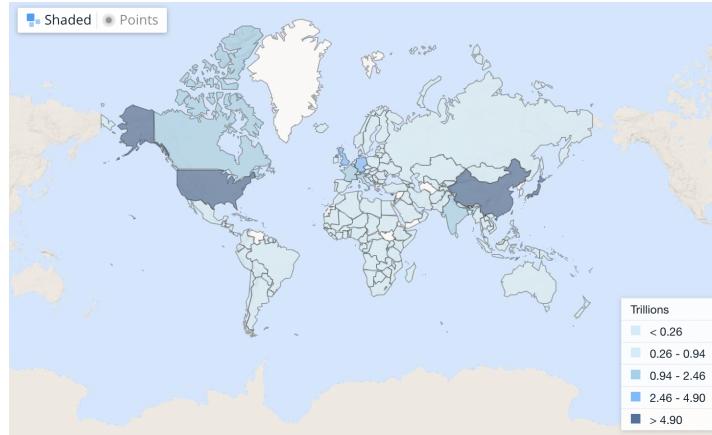


Figura 4.2: Mapa de los países según PIB de 2020. Obtenido en [2]

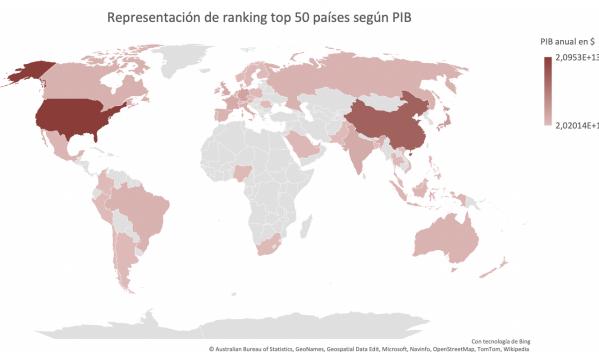


Figura 4.3: Mapa del top 50 países según su PIB en 2020.

#### 4.1.2. Importación de los datos a Python y generación de los rankings

Para este trabajo el lenguaje obtenido es Python y en concreto la herramienta Jupyter Notebooks (<https://jupyter.org/>). Para el manejo de los datos en crudo utilizaremos la librería Pandas (<https://pandas.pydata.org/>) y en concreto su estructura de datos más importante, el Dataframe. Además utilizaremos la librería numpy (<https://numpy.org/>) para la utilización de vectores y matrices.

Se explica como podemos descargar el código en el apéndice C. Lo ideal para reproducir el estudio es utilizar el fichero `.ipnyb` con jupyter notebook o google collab para poder ir visualizando los resultados paso a paso, ya que se ha desarrollado con esa intención y si se ejecuta todo el código en un solo fichero (si descargáramos el `.py`) no tendría mucha utilidad. Algunas líneas utilizadas se han escrito para que en el notebook podamos visualizar distintos elementos (estructura de variables, visualización de gráficos o dataframes...) y tomar decisiones en función de como son dichos elementos.

## 4.1. Obtención y preparación de los datos

---

En este trabajo veremos algunas porciones selectas del código para explicar el desarrollo del estudio.

Con la función de Pandas *read\_excel* pasamos a Python los datos según los teníamos en Excel para así poderlos manipular desde el código. Podemos visualizar la forma que tiene el dataframe en [4.4](#).

	Out[6]:									
	Country Name	Country Code	2001 [YR2001]	2002 [YR2002]	2003 [YR2003]	2004 [YR2004]	2005 [YR2005]	2006 [YR2006]		
0	United States	USA	1.058182e+13	1.093642e+13	1.145824e+13	1.221373e+13	1.303664e+13	1.381461e+13	1.441	
1	China	CHN	1.339396e+12	1.470550e+12	1.660288e+12	1.955347e+12	2.285966e+12	2.752132e+12	3.551	
2	Japan	JPN	4.374712e+12	4.182846e+12	4.519562e+12	4.893116e+12	4.831467e+12	4.601663e+12	4.571	
3	Germany	DEU	1.944107e+12	2.068624e+12	2.496129e+12	2.809188e+12	2.845803e+12	2.992197e+12	3.421	
4	United Kingdom	GBR	1.643908e+12	1.784077e+12	2.057094e+12	2.421814e+12	2.544829e+12	2.717060e+12	3.101	
5	India	IND	4.854410e+11	5.149379e+11	6.076993e+11	7.091485e+11	8.203816e+11	9.402599e+11	1.211	
6	France	FRA	1.376485e+12	1.494287e+12	1.840481e+12	2.115742e+12	2.196126e+12	2.318594e+12	2.651	
7	Italy	ITA	1.167013e+12	1.270712e+12	1.574146e+12	1.803227e+12	1.857524e+12	1.947920e+12	2.211	
8	Canada	CAN	7.389818e+11	7.606493e+11	8.955406e+11	1.026690e+12	1.173109e+12	1.319265e+12	1.461	
9	Korea Rep.	KOR	5.476582e+11	6.272461e+11	7.027173e+11	7.931750e+11	9.349011e+11	1.053217e+12	1.171	
10	Russian Federation	RUS	3.066021e+11	3.454705e+11	4.303478e+11	5.910167e+11	7.640171e+11	9.899305e+11	1.291	
11	Brazil	BRA	5.599837e+11	5.097953e+11	5.582337e+11	6.692893e+11	8.916338e+11	1.107627e+12	1.391	
12	Australia	AUS	3.790839e+11	3.953427e+11	4.673908e+11	6.141663e+11	6.950752e+11	7.475562e+11	8.531	
13	Spain	ESP	6.272868e+11	7.053943e+11	9.054921e+11	1.067093e+12	1.153286e+12	1.259344e+12	1.471	
14	Mexico	MEX	7.507000e+11	7.701601e+11	7.800000e+11	7.800000e+11	8.774701e+11	8.756074e+11	1.1661	

Figura 4.4: Estructura del dataframe de Pandas

Para generar los rankings seguimos el siguiente método:

- Creamos una lista de duplas donde vamos a incluir el año y la lista de los países que supondrán el ranking.
- Seleccionamos en una estructura de datos el nombre de la cabecera de la columna en cada año para poder acceder a ella con el método «loc».
- Para cada elemento en la lista de años, seleccionamos los 50 nombres ordenados por PIB de manera descendente en ese año y lo añadimos a la tupla. De esta forma tendremos el ranking de ese año.

Codificándolo en python, obtenemos el código [4.1](#). De esta forma tenemos un algoritmo genérico para generar una familia de rankings dado un dataframe obtenido del DataBase. En el código, en vez de elegir el nombre del país completo, para una mayor compactación en los datos y en su posterior representación al construir el grafo utilizamos la columna «Country code» que es el código ISO 3166-1 del país correspondiente. [\[22\]](#)

Por último, para ver de una forma más compacta, colocamos las tuplas juntas formando una matriz y al calcular la traspuesta tenemos la matriz con los rankings en los 20 años [4.5](#).

Código 4.1: Generación de los rankings

```

1 listaYears = df.columns.values[2::] #cabeceras de las columnas
   con valores para cada año
2 rankings = [] #lista de tuplas
3 for i in listaYears:
4     r = df.loc[:,["Country Code",i]]
5     rankOrdered = r.sort_values(i,ascending=False) #obtenemos
       el ranking ordenado
6     l = np.asarray(rankOrdered.iloc[:,0]) #convertimos en numpy
       array la lista de países
7     rankings.append([i[0:4],l]) #insertamos en la tupla el
       elemento [año,ranking]

```

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
1	USA																			
2	JPN	CHN																		
3	DEU	DEU	DEU	DEU	DEU	DEU	CHN	CHN	CHN	JPN										
4	GBR	GBR	GBR	GBR	GBR	CHN	DEU													
5	FRA	FRA	FRA	FRA	CHN	GBR	GBR	FRA	FRA	GBR	FRA	GBR	FRA	GBR						
6	CHN	CHN	CHN	CHN	FRA	FRA	FRA	FRA	GBR	GBR	GBR	FRA	GBR	FRA	FRA	FRA	IND	FRA	IND	IND
7	ITA	BRA	BRA	BRA	BRA	BRA	IND	IND	FRA	IND	FRA	IND	FRA							
8	MEX	MEX	ESP	ESP	CAN	CAN	ESP	BRA	BRA	ITA	ITA	RUS	RUS	ITA	ITA	BRA	ITA	ITA	ITA	ITA
9	CAN	CAN	CAN	CAN	ESP	ESP	CAN	RUS	ESP	IND	RUS	ITA	ITA	RUS	BRA	BRA	ITA	BRA	BRA	CAN
10	ESP	ESP	MEX	KOR	KOR	BRA	BRA	ESP	CAN	CAN	IND	CAN	IND	CAN	CAN	CAN	CAN	CAN	CAN	KOR
11	BRA	KOR	KOR	MEX	BRA	KOR	RUS	CAN	IND	RUS	CAN	IND	CAN	CAN	KOR	KOR	KOR	RUS	RUS	RUS
12	KOR	IND	IND	IND	MEX	RUS	IND	IND	RUS	ESP	ESP	AUS	AUS	KOR	RUS	RUS	RUS	RUS	KOR	BRA
13	IND	BRA	NLD	BRA	IND	MEX	KOR	MEX	KOR	AUS	AUS	ESP	KOR	AUS	AUS	ESP	AUS	AUS	ESP	AUS
14	NLD	NLD	BRA	NLD	RUS	IND	MEX	AUS	AUS	KOR	KOR	KOR	ESP	ESP	AUS	ESP	ESP	AUS	ESP	AUS
15	AUS	KOR	MEX																	
16	RUS	RUS	RUS	RUS	NLD	NLD	NLD	NLD	NLD	IDN	TUR	TUR	TUR	IDN						
17	CHE	CHE	CHE	TUR	TUR	TUR	TUR	TUR	TUR	IDN	TUR	IDN	NLD	IDN	TUR	TUR	NLD	NLD	NLD	NLD
18	ARG	SWE	SWE	CHE	CHE	CHE	CHE	CHE	CHE	IDN	TUR	NLD	NLD	NLD	NLD	SAU	SAU	CHE	CHE	TUR
19	SWE	BEL	BEL	SWE	SWE	SWE	SWE	POL	IDN	CHE	CHE	SAU	SAU	SAU	CHE	CHE	CHE	TUR	TUR	TUR
20	RFI	TUR	TUR	RFI	RFI	RFI	RFI	RFI	SAU	RFI	SAU	SAU	CHE	CHE	CHE	SAU	SAU	SAU	CHE	CHE

Figura 4.5: Estructura de los rankings en Python

Por último exportamos al disco con la función «to\_csv» esta estructura para que al ejecutar distintas pruebas no tengamos que volver a realizar todo el proceso sino que simplemente nos valga con cargar este archivo.

#### 4.1.3. Construcción del grafo de competitividad

Tras el anterior apartado, disponemos de la familia de rankings  $\mathcal{R}$  con la que tenemos que generar el grafo de competitividad  $GC(\mathcal{R})$ . El algoritmo que veremos será para calcular la **matriz de adyacencia** del grafo de competitividad **ponderado**. Para cuando necesitemos utilizar el no ponderado simplemente tendremos que cambiar todos los elementos de la matriz mayores que 0 por 1.

Veamos el pseudocódigo del algoritmo genérico que hemos desarrollado para la generación del Grafo de competitividad ponderado.

**Descripción del algoritmo:** Sea  $\mathcal{R} = \{c_1, c_2, \dots, c_r\}$  la familia de rankings de los nodos  $\mathcal{N} = \{n_1, \dots, n_k\}$ . Sea  $M$  la matriz de adyacencia de  $G_c(\mathcal{R})$ . Sea  $\text{pos}(c, j)$  una función que devuelve la posición del nodo  $j$  en el ranking  $c$ :

---

**Algoritmo 1** Generación del grafo de competitividad **input**=( $R, \mathcal{N}$ ) **output**= $M$

---

```

for  $i \in \{1, \dots, r - 1\}$  do
    for  $j \in \{1, 2, \dots, k - 1\}$  do
        if  $\text{pos}(c_i, n_j) > \text{pos}(c_{i+1}, n_j)$  then
            Para todo elemento  $c_e$  tal que  $\text{pos}(c_i, e) < \text{pos}(c_i, n_j)$ 
            SI  $n_j \prec_{c_i} c_e$  y  $c_e \prec_{c_{i+1}} n_j$  ( $c_e$  ha adelantado  $c_i$ )  $\rightarrow M[i, e] = M[i, e] + 1$  y  $M[e, i] = M[e, i] + 1$ 
        end if
    end for
end for

```

---

La **complejidad** en tiempo del algoritmo 1 es de  $O(r \cdot k^2)$  siendo  $r$  el número de rankings y  $k$  el número de nodos.

Explicando brevemente lo que hace el algoritmo, para cada elemento en cada ranking, comprueba si ha decrementado su posición en el ranking siguiente, y en ese caso, para todos los elementos que le han adelantado, añadimos 1 a la matriz de adyacencia. Esto se realiza para todos los rankings, del primero al penúltimo (ya que nos fijamos en el actual y en el siguiente, no podemos ir al último).

Veamos la implementación en Python del código específico para generar la matriz de adyacencia del grafo de competitividad de nuestro problema.

Primero de todo necesitamos dos estructuras de datos auxiliares para que podamos mapear de nombre de país a índice en la matriz de adyacencia y viceversa. Construimos una estructura sencilla que consiste en dos diccionarios que nos permitirán hacer este mapeo de manera eficiente en cada iteración del algoritmo de generación del grafo. Es decir, si España ocupa la posición 8 en el ranking 2, los diccionarios nos indicarán:  $\text{posicion}(\text{ESP}, 2) = 8$  y  $\text{pais\_en}(8, 2) = \text{«España»}$ .

Siguiendo la estructura del algoritmo genérico 1 y con ayuda de los diccionarios, implementamos el código para nuestro problema.

La entrada del código que veremos a continuación para generar el grafo de competitividad tiene como salida la matriz de adyacencia del Grafo de competitividad ponderado. Como entrada utilizamos la familia de rankings elaborada anteriormente y los diccionarios que ya hemos mencionado.

Código 4.2: Implementación algoritmo generación grafo de competitividad

```

1 for i in range(len(rankings)-1): #i avanza hasta el penultimo
2     rankAct = rankings[i][1]
3     rankSig = rankings[i+1][1]
4     for pais in rankAct:
5         pos_act = posicionEnRanking(rankAct, pais)
6         if(pos_act)!=len(rankAct): #no es el ultimo
7             #vemos si ha bajado
8             pos_sgt = posicionEnRanking(rankSig,pais)
9             if(pos_sgt > pos_act):
10                 #el pais ha bajado, ver los que le han
11                 #adelantado y
12                 #aadir 1 a la matriz de adyacencia
13                 for pais_inf in rankAct[pos_act+1::]:
14                     if(posicionEnRanking(rankSig,pais_inf) <
15                         pos_sgt): #uno que estaba abajo le ha
16                         #adelantado
17                         #aadir 1 a la matriz de ady:
18                         #sacamos indices
19                         ind1 = dictVertices[pais]
20                         ind2 = dictVertices[pais_inf]
21                         #sumamos a la matriz de ady
22                         MatAdy[ind1][ind2] = MatAdy[ind1][ind2]
23                             + 1
24                         MatAdy[ind2][ind1] = MatAdy[ind2][ind1]
25                             + 1
26                         #comprobacion:
27                         print(pais_inf + " ha adelantado a " +
28                             pais)
29     print() #imprimimos cada adelantamiento en forma de logs

```

En 4.6 vemos una representación del grafo ponderado. Para esta representación y las siguientes usamos la librería networkx. [19]. Esta librería tiene distintas herramientas para trabajar con grafos. En este caso usamos las herramientas de representación pero en casos posteriores utilizaremos otras implementaciones, como por ejemplo, la del algoritmo de Girvan-Newman.

Existen varias formas de representar grafos, en este trabajo utilizaremos varias. Para una primera interpretación de las conexiones es común utilizar la representación circular A.1. Sin embargo, si queremos hacer una visualización que nos permita identificar a simple vista las que en la librería networkx se denominan como kamada\_kawai o spring 4.6.

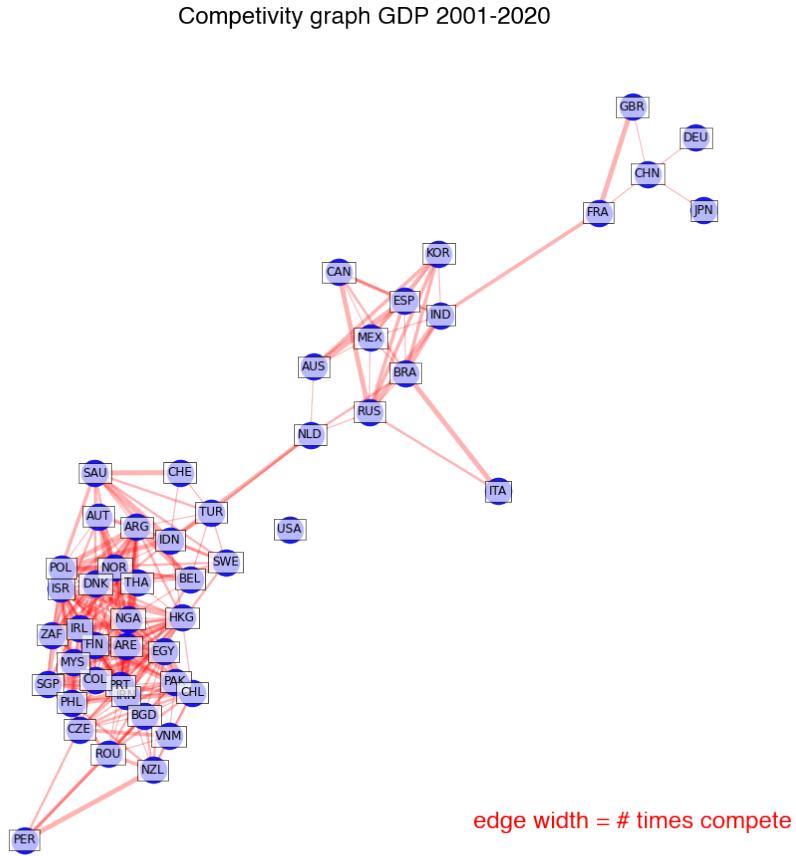


Figura 4.6: Grafo de competitividad ponderado.

## 4.2. Análisis de competitividad

En este apartado, analizaremos la competitividad de varias familias de rankings. Además de utilizar la familia de rankings principal en este trabajo (los rankings de 2001 a 2020 de los 50 países con mayor PIB en 2020) analizaremos otras familias.

Para preparar los datos y calcular sus grafos de competitividad realizaremos los mismos pasos descritos en [4.1](#) utilizando otros indicadores de la misma fuente mencionada anteriormente, los *World Development Indicators* del World Bank.

Hemos querido elegir indicadores diversos de la serie de datos "world development indicators" que pudieran representar de alguna forma el desarrollo de cada país midiendo cada uno cosas diferentes. Existen muy indicadores de naturalezas muy distintas (no todos económicos aunque sí con influencia en la economía), como el acceso a la escolarización o a la electricidad, la igualdad de género en educación universitaria o el porcentaje de población viviendo en ciudades. El

problema es que para estos indicadores no están disponibles todos los datos y encontramos numerosos «huecos» vacíos, y por tanto, sea cual sea la estrategia para llenar los valores nulos nuestros resultados van a carecer de precisión.

Por la razón anterior hemos consultado 4 valores macroeconómicos además del PIB total. En los indicadores que hemos elegido (todos de la misma base de datos citada anteriormente) solo encontramos un valor vacío en dos de las series. Este valor concretamente es el último valor (correspondiente al año 2020) de Suiza (en el estándar ISO 3166-1 usa las siglas CHE). Para llenar este valor utilizamos como estrategia usar el valor anterior, ya que consideramos que es una estrategia sencilla y que conserva la coherencia del grafo más allá que otras como llenar con la media o con un 0.

Los 4 indicadores elegidos son:

1. Export value index (índice del valor de exportaciones).
2. Import value index (índice del valor de importaciones).
3. GDP growth annual % (porcentaje anual de crecimiento de PIB).
4. GDP per capita (PIB per capita).

Para hacer la query (consulta) a la base de datos seguimos el procedimiento descrito en apartados anteriores. Al seleccionar la lista de países elegimos en todos los casos la lista de los 50 países con mayor PIB en 2020. Es decir, utilizamos **la misma lista de países** que en apartados anteriores. Hemos decidido hacerlo así para analizar siempre la competitividad entre los mismos países según diferentes indicadores macroeconómicos y poder establecer una comparación entre ellos.

Por ejemplo, en el indicador *GDP per capita* en nuestro ranking el país con mayor PIB per capita en 2020 es Suiza (siglas CHE) pero no quiere decir que sea el país con mayor PIB per cápita del mundo que en este caso es Luxemburgo, que no aparece en nuestra lista de los 50 países consultados.

Procedemos a construir el grafo de competitividad y calculamos varios indicadores que cuantifican la competitividad.

En la tabla 4.1 podemos encontrar el cálculo de los coeficientes explicados en 3.1.4 aplicado a los grafos de competitividad de nuestro indicador principal (PIB total) y al de los cuatro indicadores anteriormente mencionados.

Sean **NMD** Normalized mean degree 3.2, **NMS** Normalized Mean Strength 3.3 y **CC** Clustering coefficient 3.5.

Como podemos ver el más competitivo es el referente al indicador de porcentaje de crecimiento anual de PIB (GDP growth annual %) según los 3 coeficientes.

Tabla 4.1: Cálculo de indicadores de competitividad en grafos con distintos indicadores

Indicador	NMD	NMS	CC
GDP (PIB total)	0.21176	0.02247	0.63351
Export Value Index	0.51216	0.05435	0.75165
GDP per capita	0.15529	0.01740	0.65340
Import Value Index	0.54588	0.05916	0.75457
GDP growth annual %	0.86275	0.20567	0.94347

Podemos ver gráficamente en la representación de su grafo ponderado de competitividad 4.7 que hay una densidad de aristas mucho más grande que en nuestro grafo principal 4.6. Esto quiere decir que los nodos compiten más veces entre ellos y por tanto hay mayor competitividad, como indican los coeficientes seleccionados.

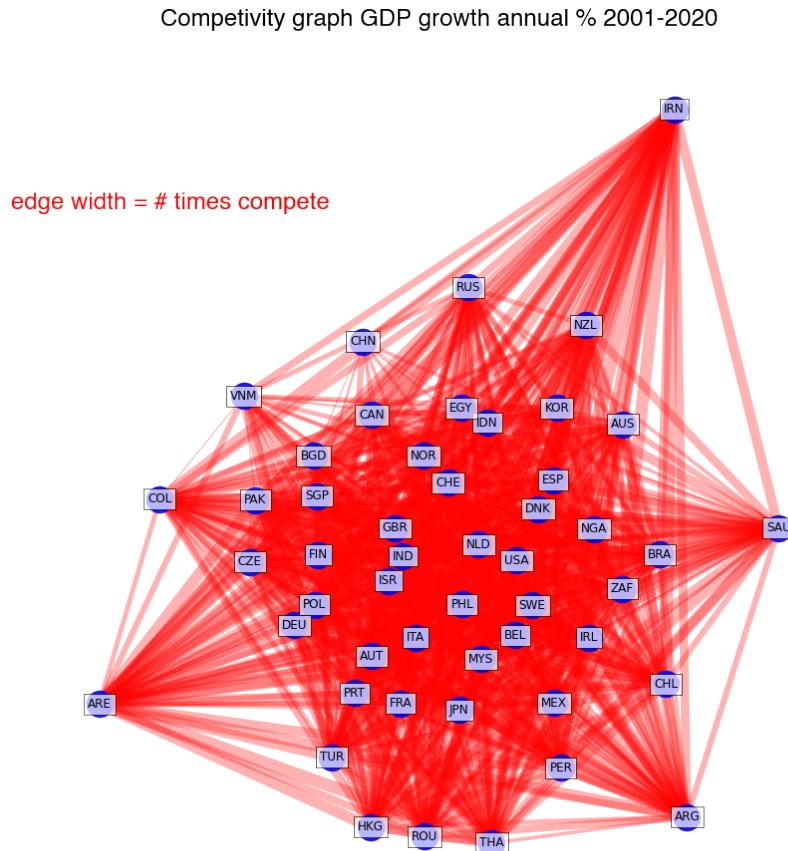


Figura 4.7: Grafo de competitividad ponderado referente al porcentaje de crecimiento anual de PIB.

En cuanto al índice que genera el grafo de competitividad con menor competitividad no podemos establecer uno con unanimidad, ya que no todos los indicadores nos dan el mismo como resultado. Segun el *NMD* y el *NMS* el indicador que genera el grafo menos competitivo es el *GDP per capita* mientras que según el *CC*, el menos competitivo es el *GDP (PIB total)*.

Mostramos la representación de los grafos estudiados en el apéndice [A.2](#).

### 4.3. Detección de comunidades

En esta sección trabajaremos con la familia de rankings principal de este problema: los rankings por PIB desde el 2001 al 2020 del top 50 países con mayor PIB en el 2020.

Realizaremos un estudio de la estructura de comunidades de esta red compleja utilizando el algoritmo Girvan-Newman que hemos detallado en [3.1.6](#). [19].

Como hemos visto en la descripción del algoritmo [3.1.6](#), el algoritmo se centra en el cálculo de la **betweenness** (betweenness de aristas). El algoritmo va eliminando en cada iteración la arista con mayor betweenness y se va repitiendo hasta que no hayan aristas restantes. Para utilizar el algoritmo de Girvan-Newman en nuestro grafo utilizaremos la implementación del algoritmo facilitada en *networkx*.

El primer paso del algoritmo es calcular la betweenness de todas las aristas de la red, para ello tenemos que usar la ecuación [3.6](#) aplicado a aristas, es decir, siendo el denominador  $b_{jk}$  el número de caminos mínimos donde interviene la arista que estamos calculando. Es decir, la betweenness de una arista mide el número de caminos mínimos del grafo en los que interfiere dicha arista.

Para calcular el camino mínimo, el algoritmo más conocido es el **algoritmo de Dijkstra**. Este algoritmo fue publicado por Edgser Dijkstra en 1959 [23] y nos da una solución sencilla y eficiente para encontrar el camino mínimo en grafos no dirigidos con ponderación positiva (si no es ponderado se asocia ponderación 1). La idea del algoritmo es explorar todos los caminos más cortos que parten de un vértice origen y que acaban en todos los demás vértices. En [2](#) vemos una implementación en pseudocódigo de este algoritmo.

Para calcular la betweenness de una arista necesitamos calcular los caminos mínimos de todo el grafo y contar el número de estos caminos en los que está presente la arista. Una vez calculada la betweenness procedemos como se detalla en [4.2](#).

En la librería *networkx* existe una implementación del algoritmo de Girvan Newmann (concretamente la función `community.girvan_newman()`). Esta nos devuelve una lista de conjuntos con las separaciones en comunidades que hace el

algoritmo de Girvan Newmann hasta haber eliminado todas las aristas.

---

**Algoritmo 2** *Algoritmo de Dijkstra* **input**=(grafo, origen).**Output**=lista de padres de cada nodo en el camino mínimo a dicho nodo desde origen

---

Sea  $w(u, v)$  la función que devuelve el peso de la arista entre los nodos u y v.

Sea  $\text{ady}(u)$  la función que devuelve la lista de adyacentes de u.

Inicializamos  $D = (\text{long}=\text{numNodos})$  con un valor infinito relativo excepto  $D[\text{origen}] = 0$  (distancias).

Inicializamos  $\text{Visitados} = \text{diccionario} (\text{long}=\text{numNodos})$  con valores 0. ( $\text{visitado}[n] = 0 \forall n$ )

Inicializamos  $\text{Padre} = \text{diccionario} (\text{long}=\text{numNodos})$  con valores 0.

$S = []$  lista vacía

Sea  $Q$  cola con valores los Nodos de G.

```

while  $Q \neq \emptyset$  do
     $u =$  nodo con min valor en  $D[u]$ 
     $S = S \cup u$ 
    for  $v \in \text{ady}(u)$  do
        if  $D[v] > D[u] + w(u, v)$  then
             $D[v] = D[u] + w(u, v)$ 
             $\text{Padre}[v] = u$ 
        end if
    end for
end while
return Padre

```

---

Los ejemplos que veremos serán con el grafo ponderado, aunque tras realizar pruebas, obtenemos exactamente los mismos resultados que si trabajamos con el grafo no ponderado.

En 4.8 podemos ver como se nos devuelve el resultado de aplicar el algoritmo.

```

({{'USA'}, {'ESP', 'FRA', 'MEX', 'ITA', 'JPN', 'IND', 'AUS', 'CHN', 'KOR', 'NLD', 'GBR', 'BRA', 'DEU', 'RUS', 'CAN'}, {'SAU', 'CHL', 'CZE', 'SGP', 'EGY', 'PER', 'IDN', 'BGD', 'NOR', 'ZAF', 'SWE', 'ISR', 'MYS', 'PRT', 'THA', 'DNK', 'IRN', 'NZL', 'IRL', 'PAK', 'HKG', 'ARG', 'POL', 'AUT', 'ARE', 'CHE', 'NGA', 'TUR', 'BEL', 'VNM', 'FIN', 'COL', 'PHL', 'ROU'}}, ({{'USA'}, {'FRA', 'JPN', 'CHN', 'GBR', 'DEU'}, {'ESP', 'MEX', 'ITA', 'IND', 'AUS', 'KOR', 'NLD', 'BRA', 'RUS', 'CAN'}, {'SAU', 'CHL', 'CZE', 'SGP', 'EGY', 'PER', 'IDN', 'BGD', 'NOR', 'ZAF', 'SWE', 'ISR', 'MYS', 'PRT', 'THA', 'DNK', 'IRN', 'NZL', 'IRL', 'PAK', 'HKG', 'ARG', 'POL', 'AUT', 'ARE', 'CHE', 'NGA', 'TUR', 'BEL', 'VNM', 'FIN', 'COL', 'PHL', 'ROU'}}, ({{'USA'}, {'FRA', 'JPN', 'CHN', 'GBR', 'DEU'}, {'ESP', 'MEX', 'ITA', 'IND', 'AUS', 'KOR', 'NLD', 'BRA', 'RUS', 'CAN'}, {'CHE', 'SAU', 'SWE', 'TUR', 'IDN'}, {'CHL', 'CZE', 'SGP', 'EGY', 'PER', 'BGD', 'NOR', 'ZAF', 'ISR', 'MYS', 'PRT', 'THA', 'DNK', 'IRN', 'NZL', 'IRL', 'PAK', 'HKG', 'ARG', 'POL', 'AUT', 'ARE', 'NGA', 'VNM', 'BEL', 'FIN', 'COL', 'PHL', 'ROU'}}, ({{'USA'}, {'FRA', 'JPN', 'CHN', 'GBR', 'DEU'}, {'ESP', 'MEX', 'ITA', 'IND', 'AUS', 'KOR', 'NLD', 'BRA', 'RUS', 'CAN'}, {'CHE', 'SAU', 'SWE', 'TUR', 'IDN'}, {'CHL', 'CZE', 'SGP', 'EGY', 'BGD', 'NOR', 'ZAF', 'ISR', 'MYS', 'PRT', 'THA', 'DNK', 'IRN', 'NZL', 'IRL', 'PAK', 'HKG', 'ARG', 'POL', 'AUT', 'ARE', 'NGA', 'VNM', 'BEL', 'FIN', 'COL', 'PHL', 'ROU'}, {'PER'}})

```

Figura 4.8: Salida del algoritmo Girvan Newman en la librería networkx

Como hemos mencionado anteriormente, el algoritmo de Girvan-Newman es jerárquico, por tanto nos permite dibujar un dendograma y establecer una estructura de comunidades «cortando» a partir de cierto nivel. En [4.9](#) mostramos el dendograma con una línea roja vertical el «corte» que hemos elegido para visualizar la separación de grupos que nos queda.

Partiendo de la salida que nos da el algoritmo con la que hemos elaborado el dendograma, podemos hacer un código que nos pinte los nodos pertenecientes a cada comunidad de un color diferente para así diferenciar claramente la estructura de comunidades.

En cuanto a qué separación elegir, es decir, en que iteración del algoritmo parar, hemos elegido la que se detalla en el dendograma, que divide el grafo en 5 comunidades. Realmente podemos separar en **no sólo** las comunidades resultantes al dividir el dendograma con una línea vertical. En cada iteración del algoritmo de Girvan-Newman se hace una división más, por tanto podríamos dividir en 4 comunidades por ejemplo (cuando esa posibilidad no existe si dividimos el dendograma con una línea vertical). Para ver una representación del grafo dividido en 4 comunidades ver [A.5](#). Esta división en 4 comunidades es la separación en comunidades que más está de acuerdo con lo que podemos identificar visualmente en una primera representación del grafo.

Existen medidas para evaluar la calidad de la estructura de comunidades de un grafo, pero no son objeto de este trabajo. En [4.10](#) vemos la representación gráfica de esta separación en estructura de comunidades sobre el grafo ponderado (mismos resultados que con el grafo no ponderado). Además, representamos con el ancho de línea la betweenness de cada arista, siendo mayor cuanto mayor betweenness tiene.

Representamos la betweenness para que veamos gráficamente las aristas que primero se eliminan. Esto solo es válido para la primera arista con mayor betweenness (la arista entre «NLD» y «IDN» tiene una betweenness de 0.332 calculada en el código con la función de *networkx edge\_betweenness\_centrality*). Tras eliminar esta arista, según el algoritmo, tenemos que recalcular la betweenness de nuevo.

En esta figura **hemos omitido «USA»** ya que es una componente inconexa y siempre va a formar una comunidad aparte y para mostrar la betweennes como pretendíamos nos era imposible de calcular añadiendo componentes inconexas. En [4.8](#) vemos que desde la primera iteración tiene una comunidad propia y en [4.6](#) comprobamos como «USA» no tiene ninguna arista incidente.

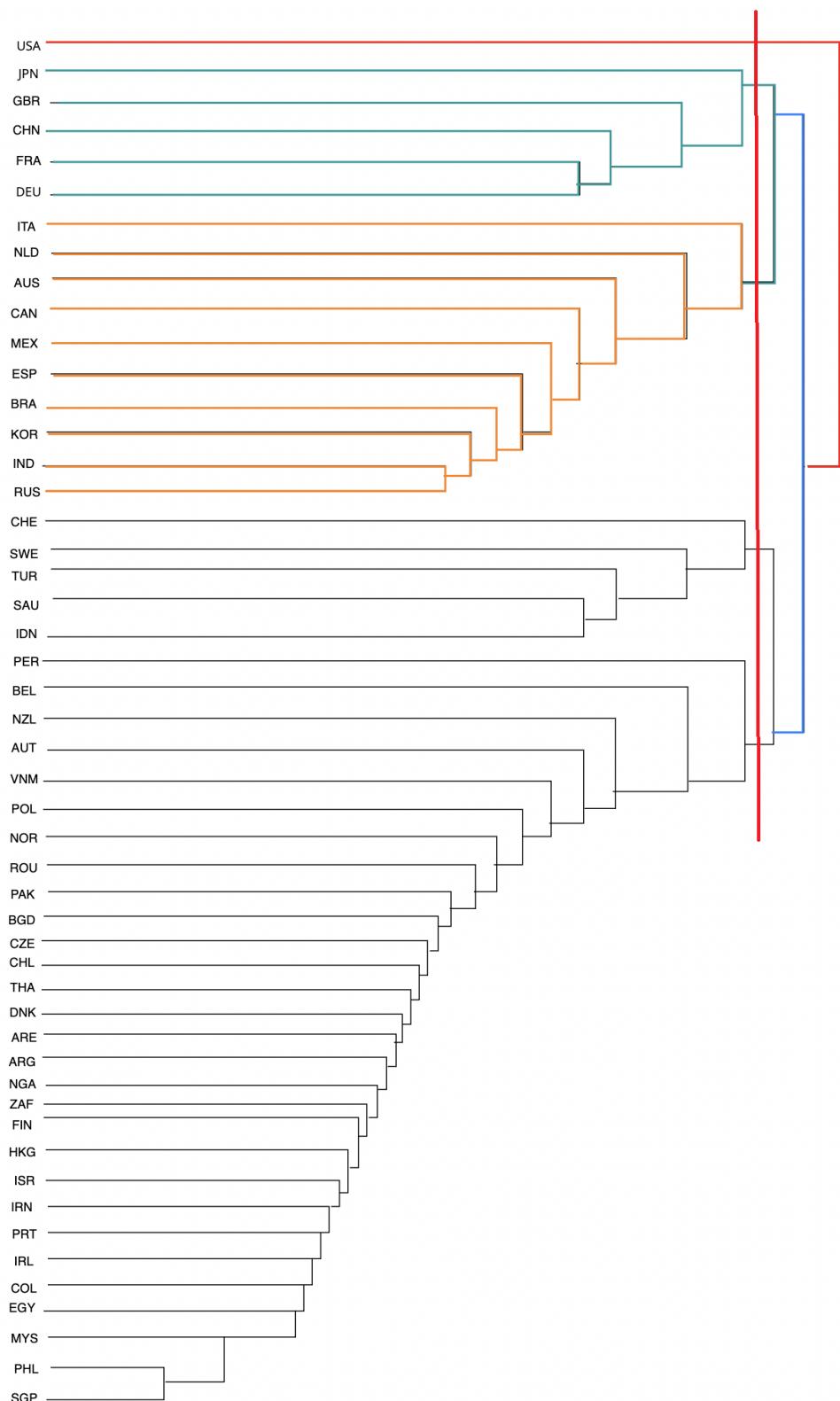


Figura 4.9: Dendograma a partir de la salida del algoritmo Girvan Newman con línea para partición de comunidades.

**Evolutive competitiveness graph 2001-2020 GDP - Only connected components(missing USA)**

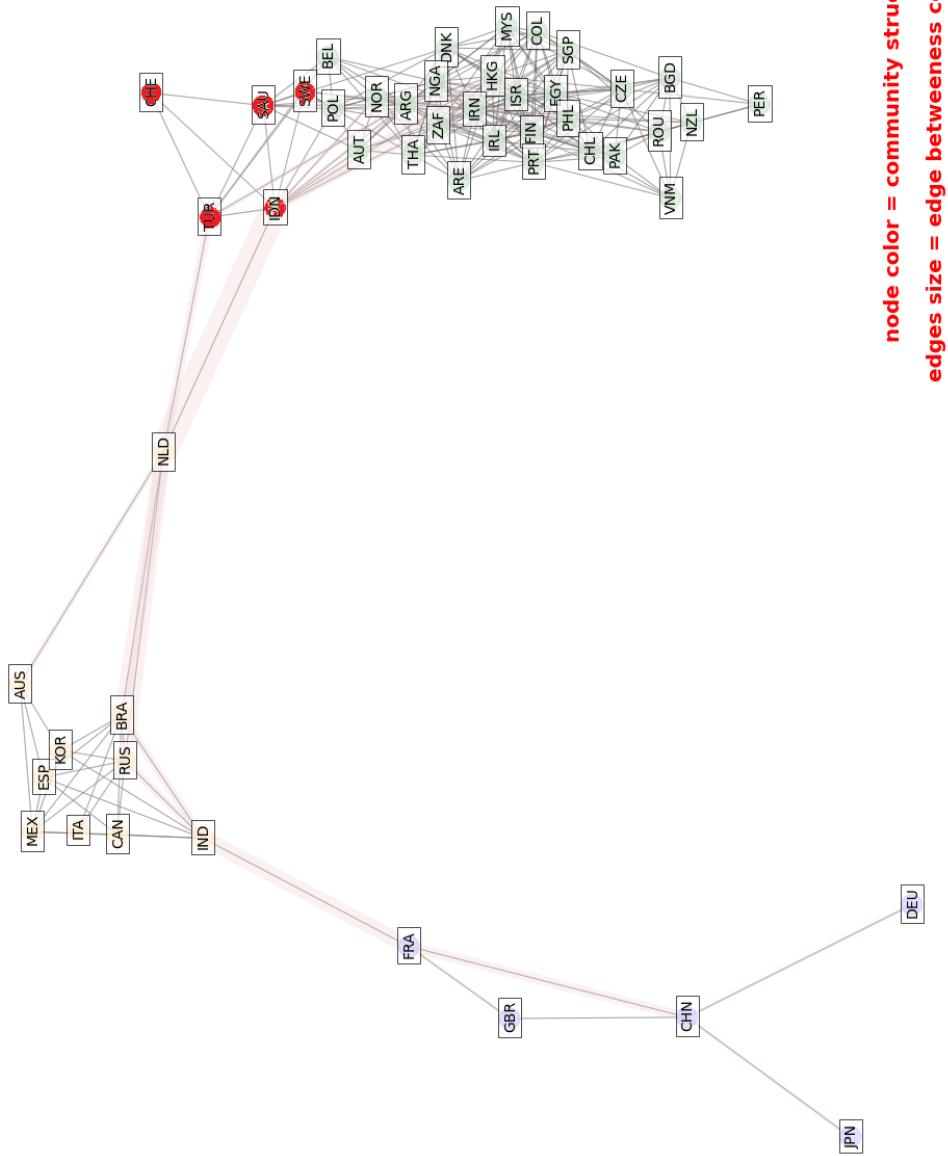


Figura 4.10: Grafo de competitividad ponderado coloreado según detección de comunidades usando algoritmo Girvan Newman.  
El ancho de cada arista es en función de su betweenness.

# 5

## Conclusiones y trabajos futuros

En este capítulo se detallan las conclusiones derivadas del TFG y la propuesta de posibles trabajos futuros.

### 5.1. Conclusiones

Como hemos podido ver en el trabajo, la estructura de grafos de competitividad es una herramienta muy potente para estudiar la competitividad en distintas familias de rankings.

Esta estructura nos ha permitido estudiar la competitividad según varios indicadores y determinar una estructura de comunidades que nos da agrupaciones de países que se han comportado de maneras similares en cuanto a su desarrollo y evolución económica medida según un indicador macroeconómico representativo.

Podemos obtener varias conclusiones claras de este trabajo.

En cuanto al apartado de **análisis de competitividad**, podemos observar que el indicador que nos induce una mayor competitividad entre países en el siglo XXI es el porcentaje de crecimiento anual de PIB (GDP growth annual %). Esta afirmación la fundamentamos en los 3 coeficientes estudiados y en la mayor densidad de aristas que observamos en este grafo frente a los otros 4 grafos estudiados generados con otros indicadores macroeconómicos. Esto puede significar que el crecimiento económico del país en bienes y servicios no ha sido siempre constante para todos los países, y que cada año el crecimiento económico del país

puede variar de manera sustancial en cuanto al porcentaje que crece. Esta afirmación acompañada de que el PIB (GDP) total del país según el *Coeficiente de clusterización* es el menos competitivo (y según el resto de coeficientes es de los menos competitivos), nos puede indicar que aunque el crecimiento porcentual de cada país no haya sido fijo en el siglo XX, su posición en cuanto al PIB general está mayormente determinada por condiciones no sólamente económicas, sino características inherentes al país como el tamaño, la industria, o el volumen de importaciones y exportaciones. Es decir, aunque el porcentaje de crecimiento de la economía de los países vaya variando, en estos años ha habido (como Estados Unidos) o han surgido (como es el caso de China) varias potencias que resultan de momento inalcanzables en cuanto a la riqueza total evaluada con el PIB del país.

Por otra parte, en cuanto al estudio de **detección de comunidades** podemos extraer varias conclusiones (Recordemos que este apartado se ha realizado tomando en cuenta únicamente el PIB anual del país). La primera y más clara es que al ser Estados Unidos una componente inconexa en los grafos y ser la número 1 en cuanto a este indicador en todos los rankings de los años estudiados (2001-2020), forma una comunidad aparte, en la que se entiende que es un país con características únicas en este aspecto y no ha competido con ningún otro país por conseguir ese primer puesto.

Por otra parte, en los países restantes, existe una mayor competitividad entre los países más pobres (viendo mayor número de aristas en el subgrafo compuesto por esta comunidad) que entre los más ricos según podemos observar en los grafos. Aparentemente (y sin haber realizado un estudio de calidad de comunidades que lo sostente) las mejores divisiones en comunidades nos separan al grafo en 4 o en 5 comunidades, estando España en ambas divisiones en el tercer grupo, por debajo de JPN, DEU, CHN, GBR y FRA y compartiendo grupo con IND, CAN, ITA, MEX, KOR, RUS, BRA, AUS y NLD (estos grupos son válidos tanto para la separación en 4 como en 5 comunidades).

## 5.2. Trabajos futuros

Proponemos diversos trabajos futuros referentes a este estudio que nos podrían dar nuevas perspectivas y una visión más clara del problema que estamos estudiando, que es la clasificación y competitividad de países.

- Evaluar la calidad de las distintas divisiones del grafo en estructuras de comunidades y establecer una división clara entre países.
- Realizar detección de comunidades en el resto de grafos estudiados además del PIB y comparar las comunidades detectadas.

- Utilizar otros indicadores que evalúen la competitividad para determinar también de una forma más precisa cuál es el índice que induce un grafo menos competitivo.
- Utilizar otros algoritmos además del algoritmo de Girvan-Newman para la detección de comunidades. Comparar con los resultados obtenidos.



# 6

## Organización del tiempo

En este apartado veremos como hemos ido realizando la organización del tiempo.

En [B](#) podemos encontrar una tabla detallada de cada sesión de trabajo dedicada para este trabajo. En total se han realizado **156 horas y 14 minutos** de trabajo hasta el momento de finalización de esta memoria. Restando aún el trabajo referente a la preparación de la presentación y defensa del trabajo.

En una primera aproximación se pretendía separar las horas de trabajo entre búsqueda de información, desarrollo de código y trabajo en la memoria. Esto ha sido imposible ya que en la mayoría de sesiones de trabajo se hacían las 3 tareas de manera simultánea.

Por último, comentar que como se puede ver en la tabla existe un parón entre el 10 y el 30 de mayo. Esto se debe a una operación por una lesión deportiva que me hizo estar parado estos días y tras poder incorporarme haber tenido que empezar a trabajar en sesiones más cortas o con más pausas.



# Bibliografía

- [1] R. Criado, E. García, F. Pedroche, and M. Romance, “Comparing rankings by means of competitiveness graphs: structural properties and computation,” 10 2013.
- [2] W. Bank. (2022) World development indicators. Se consultan los indicadores especificados en el texto entre los años 2001 y 2020. [Online]. Available: <https://databank.worldbank.org/source/world-development-indicators>
- [3] R. Criado, E. García, F. Pedroche, and M. Romance, “A new method for comparing rankings through complex networks: Model and analysis of competitiveness of major european soccer leagues,” *Chaos (Woodbury, N.Y.)*, vol. 23, p. 043114, 12 2013.
- [4] M. Girvan and M. Newman, “Girvan, m. newman, m. e. j. community structure in social and biological networks. proc. natl acad. sci. usa 99, 7821-7826,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 7821–6, 07 2002.
- [5] GeeksForGeeks, “Clustering coefficient in graph theory – geeks for geeks,” 2022, [Internet; descargado 10-junio-2022]. [Online]. Available: <https://www.geeksforgeeks.org/clustering-coefficient-graph-theory/>
- [6] H. Mahyar, H. Rabiee, A. Movaghar, E. Ghalebi, and A. Nazemian, “Cs-comdet: A compressive sensing approach for inter-community detection in social networks,” pp. 89–96, 08 2015.
- [7] M. Zarei, K. Aghababaei Samani, and G. Omidi, “Complex eigenvectors of network matrices give better insight into the community structure,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, p. P10018, 10 2009.
- [8] A. McLaughlin and D. Bader, “Scalable and high performance betweenness centrality on the gpu,” *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, vol. 2015, pp. 572–583, 01 2015.
- [9] B. Rees and K. Gallagher, *EgoClustering: Overlapping Community Detection via Merged Friendship-Groups*, 12 2013, vol. 6, pp. 1–20.
- [10] Wikipedia, “Producto interno bruto — wikipedia, la enciclopedia libre,” 2022, [Internet; descargado 27-junio-2022]. [Online]. Available: [https://es.wikipedia.org/w/index.php?title=Producto\\_interno\\_bruto&oldid=144228037](https://es.wikipedia.org/w/index.php?title=Producto_interno_bruto&oldid=144228037)
- [11] R. Criado and R. Muñoz, “Un semestre de matemática discreta,” 2007.
- [12] I. Gribkovskaia, Halskau Sr., and G. Laporte, “The bridges of königsberg—a historical perspective,” *Networks*, vol. 49, no. 3, pp. 199–203, 2007.
- [13] Wikipedia, “Leonhard euler — wikipedia, la enciclopedia libre,” 2022, [Internet; descargado 27-junio-2022]. [Online]. Available: [https://es.wikipedia.org/w/index.php?title=Leonhard\\_Euler&oldid=144165240](https://es.wikipedia.org/w/index.php?title=Leonhard_Euler&oldid=144165240)

## BIBLIOGRAFÍA

---

- [14] P. Erdős and A. Rényi, “On random graphs i,” *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959. [Online]. Available: <https://www.bibsonomy.org/bibtex/2420b83c1533188c0b54bd1f6eea2b782/krevelen>
- [15] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [16] Y. Li and H. Wu, “A clustering method based on k-means algorithm,” *Physics Procedia*, vol. 25, pp. 1104–1109, 12 2012.
- [17] M. Newman, “Detecting community structure in networks,” *Eur Phys J*, vol. 38, 03 2004.
- [18] ———, “Fast algorithm for detecting community structure in networks. phys. rev. e stat. nonlin. soft. matter. phys. 69(6 pt 2), 066133,” *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 69, p. 066133, 07 2004.
- [19] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using networkx,” in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 – 15.
- [20] Economipedia, “Producto interior bruto – economipedia,” 2022, [Internet; visto por última vez 22-junio-2022]. [Online]. Available: <https://economipedia.com/definiciones/producto-interior-bruto-pib.html>
- [21] E. Mundo, “Hugo chávez anuncia la salida de venezuela del banco mundial y el fmi,” 2007, [Artículo correspondiente a el Mundo internacional del día 1 de Mayo de 2007. Accedido digitalmente]. [Online]. Available: <https://www.elmundo.es/elmundo/2007/05/01/internacional/1177983208.html>
- [22] O. I. para la Normalización (ISO), “Iso 3166,” 1974. [Online]. Available: <https://www.iso.org/iso-3166-country-codes.html>
- [23] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 1, no. 1, p. 269–271, dec 1959. [Online]. Available: <https://doi.org/10.1007/BF01386390>



# **Apéndices**



A

# Representaciones del grafo

En esta sección veremos varias imágenes referentes a representaciones de los distintos grafos utilizados y referenciados a lo largo del texto. 4.

### A.1. Representación del grafo general no ponderado

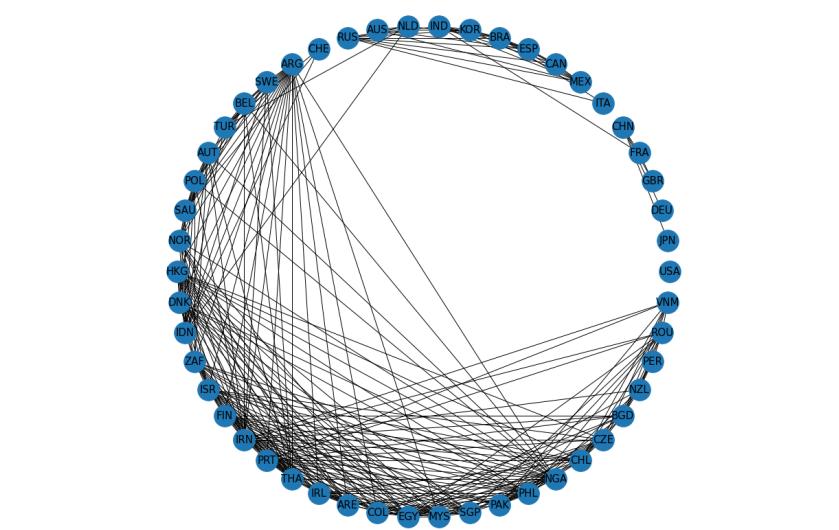


Figura A.1: Grafo de competitividad no ponderado por PIB

## A.2. Representaciones de los grafos según los indicadores mencionados en 4.2

Competitvity graph Exports value index 2001-2020

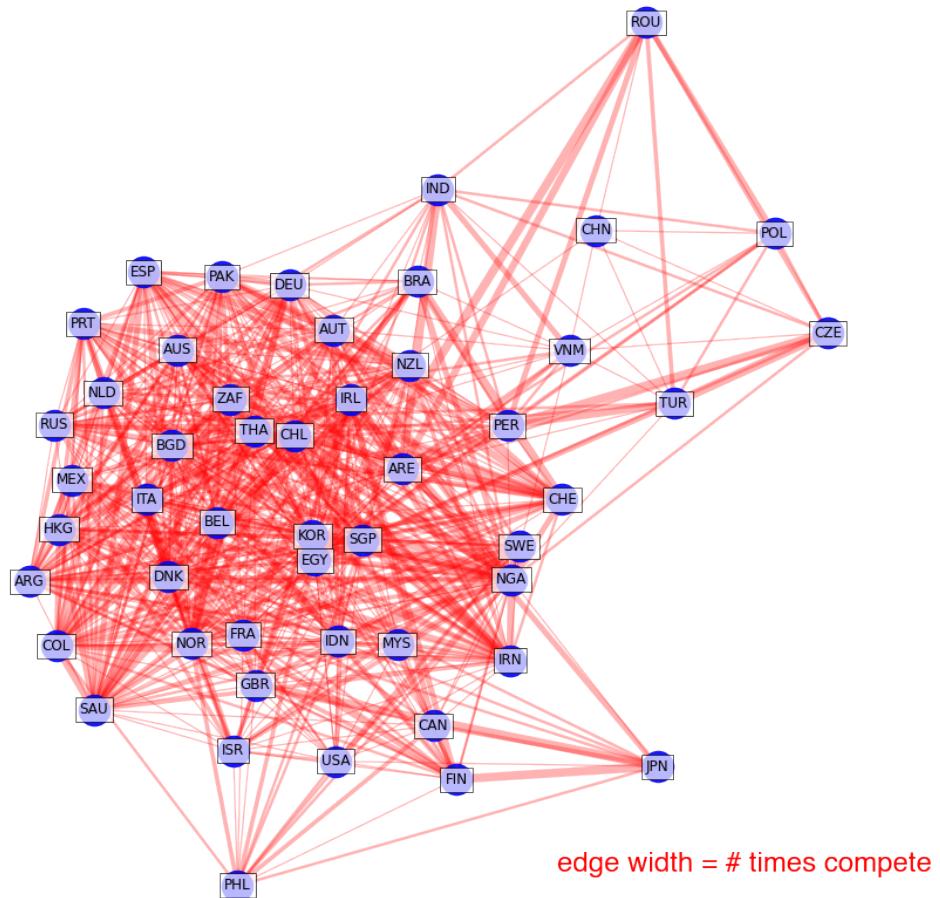


Figura A.2: Grafo de competitividad ponderado referente al índice de exportaciones

Competitvity graph Import Value Index 2001-2020

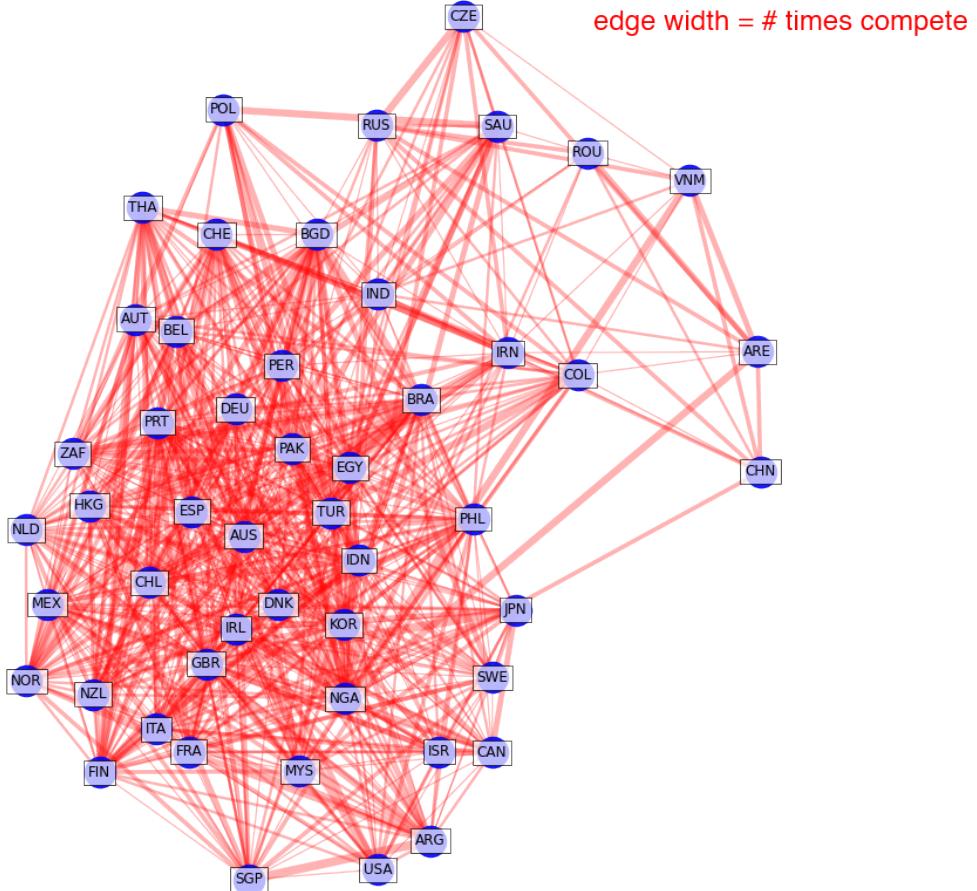


Figura A.3: Grafo de competitividad ponderado referente al índice de importaciones

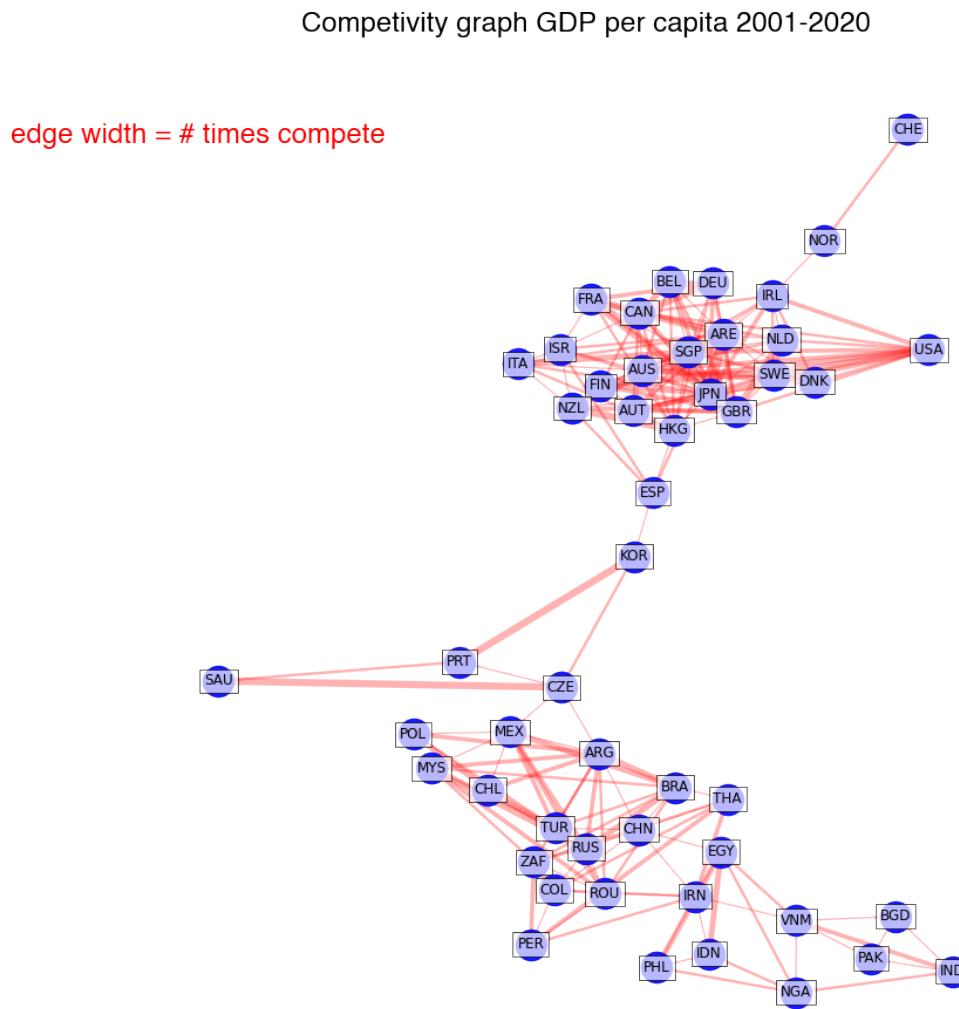


Figura A.4: Grafo de competitividad ponderado referente al GDP (PIB) per capita

### A.3. Representaciones referenciadas en 4.3

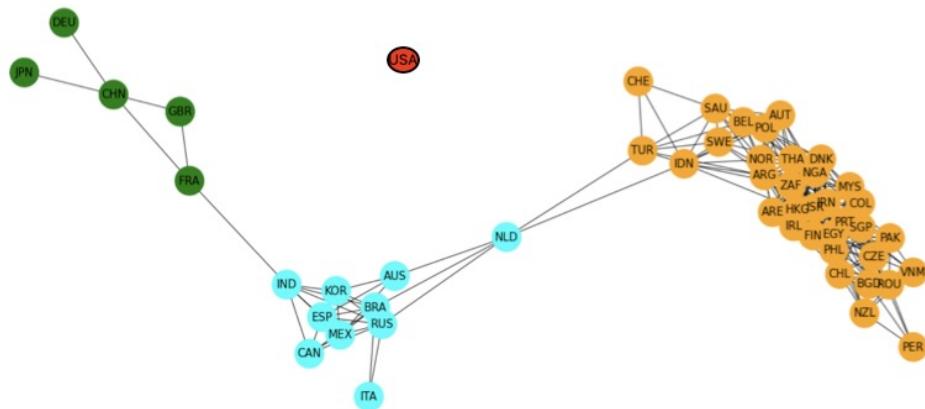


Figura A.5: Grafo no ponderado con detección de comunidades

# B

Tabla organización del tiempo

## Capítulo B. Tabla organización del tiempo

---

### Organización del tiempo para TFG Matemáticas

Fecha	Hora inicio	hora fin	Duración	Descripción de tarea
23/2/22	10:00	11:30	1:30	Lectura de documentación (TFM detección comunidades)
23/2/22	12:00	13:15	1:15	Lectura de documentación (Intentional Risk Management through Complex Networks Analysis punto 2.2)
24/2/22	11:00	11:27	0:27	Lectura de documentación (Intentional Risk Management through Complex Networks Analysis punto 2.2)
24/2/22	11:30	12:30	1:00	Lectura de documentación (paper de competitividad teórico)
25/2/22	8:45	10:15	1:30	Lectura de documentación (paper de competitividad teórico + práctico)
26/2/22	10:00	11:20	1:20	Lectura de documentación (Repaso teoría de grafos)
1/3/22	11:45	14:00	2:15	Búsqueda de información sobre PIBs y consultas al BM y edición plantilla Latex
1/3/22	17:30	19:45	2:15	Preparación de los datos, limpieza y código de pruebas en python para preparar los rankings
2/3/22	11:30	13:00	1:30	Continuación pruebas código. Lectura documentación librería de grafos networkx.
5/3/22	10:00	12:30	2:30	Continuación apartado 3, definición grafos hasta grafos de competitividad
9/3/22	10:00	13:00	3:00	Definición grafos competitividad + repaso papers
10/3/22	10:00	12:45	2:45	Algoritmos preparación de datos a rankings
13/3/22	9:30	13:15	3:45	Preparación de rankings en matrices. Repaso librerías numpy y pandas
14/3/22	10:00	13:00	3:00	Repaso np y pandas, continuación anterior
15/3/22	10:00	13:00	3:00	Código para generar rankings
16/3/22	10:00	13:00	3:00	Limpiar código anterior y empezar el código de GC
17/3/22	10:00	13:00	3:00	Memoria apartado rankings y GC
21/3/22	9:00	11:00	2:00	Continuación apartado rankings y GC
22/3/22	9:30	11:00	1:30	Empezar algoritmo GC, pensar como hacerlo y buscar info
23/3/22	9:30	11:30	2:00	No encuentro otra implementación, hacerla de 0
24/3/22	9:30	12:00	2:30	Continuar con desarrollo algoritmo GC
25/3/22	9:30	12:30	3:00	continuación GC + pruebas con grafos simples
28/3/22	9:30	12:00	2:30	Redacción memoria (resolver problemas con latex)
29/3/22	10:00	12:30	2:30	Desarrollo del código para generar los grafos II + visualización grafos
17/4/22	11:00	13:30	2:30	empezar a dibujar grafos + documentación lib networkx
18/4/22	11:00	14:00	3:00	Ver formas de representar grafos como en paper
19/4/22	9:00	12:30	3:30	Lectura documentación paper futbol
22/4/22	10:00	11:30	1:30	Cuantificación de competitividad documentación
25/4/22	9:15	13:00	3:45	Consulta duda a Regino, releer papers para aplicar procesos, cálculo de coeficientes... + continuación redacción memoria
27/4/22	9:30	11:30	2:00	Repasar coeficientes para grafos ponderados y no ponderados, empezar redaccion memoria esa parte
28/4/22	9:30	11:30	2:00	Empezar redaccion memoria parte resultados
5-may	10:30	12:00	1:30	Documentación en memoria parte de datos a datos preparados
6/5/22	10:00	12:30	2:30	Continuación memoria
8/5/22	11:30	12:30	1:00	Continuación memoria
9/5/22	11:00	13:00	2:00	Continuar parte de algoritmo de GC en la memoria + explicacion
30/5/22	16:00	19:00	3:00	Repaso TFG y memoria, ver dudas para reunión mañana. Corrección cosas y estilo código. Lento por operación
1/6/22	10:00	12:15	2:15	Estudio de algoritmo GN y empiezo a desarrollar código para detección com
1/6/22	17:30	19:15	1:45	Continuación código detección com
2/6/22	10:00	12:45	2:45	Búsqueda info sobre visualizaciones networkx y como geneerar dendograma
6/6/22	16:30	19:30	3:00	Como hacer dendrograma pruebas de codigo
7/6/22	16:00	18:00	2:00	Continuación arriba
8/6/22	10:00	12:30	2:30	Dendrograma a mano
8/6/22	16:30	19:00	2:30	Dendrograma a mano
9/6/22	10:30	13:00	2:30	Memoria apartado detección com
11/6/22	10:00	12:45	2:45	Continuar memoria apartado det com
13/6/22	10:00	13:00	3:00	Parte memoria ecuaciones coef cuant
14/6/22	10:00	13:00	3:00	Mejora memoria apartado det com
15/6/22	10:00	13:00	3:00	Generación de imágenes de otras formas para menorar
17/6/22	10:00	13:00	3:00	Continuar memoria apartado det com
20/6/22	10:00	13:00	3:00	Aplicación cuant comp a grafo
21/6/22	10:00	13:00	3:00	Continuación arriba
22/6/22	10:00	13:00	3:00	Memoria cuant. Comp
23/6/22	10:00	13:00	3:00	Mejora diagramas de detección de com + dendrograma cortado. Editar fotos
24/6/22	11:00	12:00	1:00	Reunión con Regino
24/6/22	16:30	20:00	3:30	Redacción memoria (introducción + ir ajustando los párrafos hasta apartado 3)
25/6/22	10:00	13:30	3:30	Código para generar imágenes que necesito. Continuación con la memoria hasta generación rankings
25/6/22	21:30	23:59	2:29	Continuación memoria hasta generación grafo de competitividad. Reorganización código
26/6/22	0:00	1:45	1:45	Continuación memoria hasta generación grafo de competitividad. Reorganización código
26/6/22	17:30	21:00	3:30	Memoria hasta apartado análisis de competitividad. Generar imágenes de grafos de todos indicadores
27/6/22	11:00	14:30	3:30	Corregir error con imágenes de arriba. Continuar memoria. Búsqueda info solucionar posicionamiento latex.
27/6/22	18:30	23:00	4:30	Continuar ajustando la memoria hasta detección de comunidades. Organización de apéndices
28/6/22	12:00	13:30	1:30	Repaso final memoria + modificar apéndices y organizar código para pegar en memoria
28/6/22	17:30	19:30	2:00	Repaso final memoria + apartado de gestión de tiempo (apartado 6).

Horas totales 156h 14min Obj >=150

Figura B.1: Tabla de organización del tiempo para este trabajo

# C

## Código utilizado

En este apéndice veremos donde podemos encontrar el código utilizado.

En este trabajo hemos usado jupyter notebook y el código se ha desarrollado para poder ir viendo los resultados en cada celda, por tanto, si queremos recrear el estudio es necesario que utilicemos el fichero .ipnyb para abrir directamente en un notebook. Podemos acceder a este fichero directamente en [https://github.com/MarioCanales/TFG\\_Matematicas](https://github.com/MarioCanales/TFG_Matematicas). Accediendo a este repositorio tendremos disponibles de una manera sencilla el código el y los distintos ficheros con los datos que se utilizan:

## Capítulo C. Código utilizado

---

MarioCanales Add files via upload		
		526f441 10 minutes ago ⏲ 4 commits
📄	README.md	Update README.md 11 minutes ago
📄	Top 50 Commercial service imports...	Add files via upload 10 minutes ago
📄	Top 50 Expense perc of GDP.xlsx	Add files via upload 10 minutes ago
📄	Top 50 Export value index .xlsx	Add files via upload 10 minutes ago
📄	Top 50 GDP growth annual perc .xlsx	Add files via upload 10 minutes ago
📄	Top 50 GDP per capita.xlsx	Add files via upload 10 minutes ago
📄	Top 50 Import value index .xlsx	Add files via upload 10 minutes ago
📄	Top 50 Service imports.xlsx	Add files via upload 10 minutes ago
📄	codeTFGMate.ipynb	Add files via upload ← 14 minutes ago
📄	top 50 Commercial Services Export...	Add files via upload 10 minutes ago
📄	top 50 hist 20 años.csv	Add files via upload 10 minutes ago

Figura C.1: Repositorio de github con fichero .ipynb con el código del estudio

Por último, recordar que las rutas que se utilizan de lectura y escritura son relativas a donde estaban guardados en mi ordenador los archivos. Una vez se descargue el código y los ficheros se deberá modificar de manera consecuente para la lectura correcta.