Cálculo de familias coespectrales de gráficas usando programación concurrente

Computation of Cospectral Graph Families Using Concurrent Programming

Luis Daniel Ramos López

Universidad Autónoma Metropolitana – Unidad Iztapalapa
Av. Ferrocarril San Rafael Atlixco, Núm. 186, Col. Leyes de Reforma 1 A Sección, Alcaldía Iztapalapa, 09310, Ciudad de México, México
*daniel.rms.lpz@outlook.com

PALABRAS CLAVE:

Método numérico, grafica, espectro, valores propios

RESUMEN:

Dos o más gráficas no isomorfas que comparten los mismos valores propios conforman una familia coespectral. Estas familias constituyen un objeto fundamental en la teoría de gráficas y son ampliamente investigadas por la información estructural que puede deducirse a partir de su espectro. En este trabajo se calcularon todas las familias coespectrales de gráficas no isomorfas de hasta 11 vértices, empleando el algoritmo de Samuelson–Berkowitz para el cálculo exacto del polinomio característico y el método de Jacobi para el cálculo numérico de valores propios, junto con técnicas de programación concurrente para procesar grandes volúmenes de gráficas. Se obtuvo la clasificación completa de las familias coespectrales hasta 10 vértices y se diseñó una arquitectura concurrente capaz de escalar el procesamiento al caso de 11 vértices, que comprende más de mil millones de gráficas no isomorfas. La infraestructura desarrollada establece un marco reproducible y escalable para estudios espectrales de mayor escala en teoría de gráficas.

KEYWORDS:

Numerical method, graph, spectra, eigenvalues

ABSTRACT:

Two or more non-isomorphic graphs sharing the same eigenvalues form a cospectral family. These families constitute a fundamental object in graph theory and are widely studied due to the structural information that can be inferred from their spectra. In this work, all cospectral families of non-isomorphic graphs with up to 11 vertices were computed using the Samuelson–Berkowitz algorithm for the exact computation of the characteristic polynomial and the Jacobi method for the numerical calculation of eigenvalues, together with concurrent programming techniques to process large volumes of graphs. A complete classification of cospectral families up to 10 vertices was obtained, and a concurrent architecture capable of scaling the computation to the case of 11 vertices—comprising more than one billion non-isomorphic graphs—was designed. The developed infrastructure establishes a reproducible and scalable framework for large-scale spectral studies in graph theory..

• Recibido: 1 de noviembre de 2025 • Aceptado: 12 de marzo de 2026 • Publicado en línea: 2 de octubre de 2026

1. INTRODUCCIÓN

Uno de los invariantes algebraicos más importantes de una gráfica es el polinomio característico de alguna de sus matrices asociadas, como la matriz de adyacencia o la

Copyright:© 2026 *por los autores*
Creative Commons Atribución 4.0 CC-BY

matriz laplaciana. Este polinomio y sus raíces, los valores propios de la matriz, constituyen el espectro de la gráfica, el cual describe ciertas características estructurales de la gráfica según el tipo de matriz considerada.

Cuando dos o más gráficas no isomorfas tienen el mismo polinomio característico se dice que forman una familia coespectral, pues comparten exactamente los mismos valores propios y, en consecuencia, comparten las propiedades estructurales que pueden deducirse a partir del espectro considerado. El estudio de las familias coespectrales es relevante porque permite entender hasta qué punto el espectro puede caracterizar a una gráfica y, al mismo tiempo, identificar aquellas que son determinadas por su espectro.

Aunque existen gráficas determinadas por su espectro, se sabe que para órdenes suficientemente grandes la mayoría de las gráficas no lo están [6], lo que hace particularmente relevante el estudio sistemático de las familias coespectrales.

Encontrar todas las familias coespectrales de gráficas con un número dado de vértices implica un proceso exhaustivo, pues se necesita calcular el polinomio característico de cada gráfica no isomorfa y agruparlas en clases según dicho polinomio. En este trabajo se aborda este problema para todas las gráficas no isomorfas de hasta 11 vértices. Para ello se emplean métodos numéricos modernos, en particular el algoritmo de Samuelson–Berkowitz para el cálculo del polinomio característico, lo que permite organizar las familias de manera exacta, y el método de Jacobi para valores propios, con el fin de analizar las características espectrales de las familias.

La implementación, desarrollada en C++ utilizando la biblioteca estándar y con técnicas de programación concurrente para el manejo eficiente de hilos a nivel de usuario, permitió procesar grandes volúmenes de gráficas y obtener de manera sistemática las familias coespectrales correspondientes. A partir de esta información se procedió a organizar y analizar los resultados para las gráficas de hasta $n=10$ vértices, con el fin de resaltar las propiedades más relevantes de las familias y las relaciones que surgieron al considerar espectros de adyacencia y laplacianos. Finalmente, en la discusión se reflexionó sobre las limitaciones impuestas por el hardware utilizado y se señalaron posibles estrategias para optimizar el cálculo, con la intención de abrir una ventana hacia futuros esfuerzos orientados a extender este tipo de análisis a gráficas con un mayor número de vértices.

2. PRELIMINARES

2.1. Definiciones fundamentales de gráficas

Las definiciones de esta subsección fueron tomadas de Chacón [1].

Sea una gráfica $G=(V,E)$, donde V es el conjunto finito de vértices de G y E es su multiconjunto de aristas. La matriz de adyacencia de G es una matriz $A=[a_{ij}]_{|v|\times|v|}$, en donde a_{ij} es el número de aristas que unen al vértice v_i con el vértice v_j.

Para cada vértice $v \in V$ se define el grado de v, denotado como $gr(v)$, como el número de aristas de G que son incidentes en él.

La matriz de grados de la gráfica es una matriz $D=[d_{ij}]_{|V|\times|V|}$ donde d_{ij} se define como sigue:

$$d_{ij}=\begin{cases} gr(v) & \text{si } i=j \\ 0 & \text{si } i\neq j \end{cases}$$

La matriz laplaciana de la gráfica es una matriz L como sigue:

$$L=D-A$$

Cuando G es una gráfica simple, la matriz de adyacencia tiene ceros en la diagonal y solo ceros y unos en las demás entradas. Como consecuencia, tanto A como L tienen entradas enteras, de modo que pueden considerarse matrices sobre el anillo de los enteros Z.

2.2 Definición del espectro de una gráfica y familias coespectrales

Sea $M \in [K]_{n\times n}$ una matriz con entradas en un anillo conmutativo K. Se define su polinomio característico como:

$$p_M(\lambda) = det(M - \lambda I_n)$$

Las raíces de $p_M(\lambda)$ se denotan como los valores propios de M. Si M es simétrica, como es el caso de la matriz de adyacencia o la laplaciana de una gráfica, todos los valores propios de M son reales [2].

Sea G una gráfica y A su matriz de adyacencia, los valores propios de A constituyen el espectro de adyacencia de G. Algunas características estructurales que el espectro de adyacencia de G nos dice son las siguientes [3]:

- El número de valores propios iguales a cero indica la dimensión del núcleo de A, que se relaciona con la existencia de vértices aislados.
- El mayor valor propio de A da información sobre el grado máximo de la gráfica y su regularidad (si la gráfica es d-regular, entonces d es un valor propio).
- Si el espectro de adyacencia es simétrico respecto al origen, la gráfica es bipartita. Además, si una gráfica es bipartita, todos los coeficientes impares de su polinomio característico deben ser 0.
- El número total de vértices es igual al grado del polinomio característico, y la suma de los valores propios es cero (porque $tr(A)=0$ en gráficas simples).

Sea L la matriz laplaciana de G, los valores propios de L constituyen el espectro laplaciano de G. Algunas características estructurales que el espectro laplaciano de G nos dice son las siguientes [4]:

- Todos los valores propios de L son no negativos.
- El número de ceros en el espectro laplaciano coincide con el número de componentes conexas de la gráfica.
- El mayor valor propio de L da información sobre el tamaño y la conectividad de la gráfica.
- La suma de los valores propios del laplaciano es igual a dos veces el número de aristas ($tr(L)=2|E|$).

Sean G y H dos gráficas no isomorfas y sus polinomios característicos $p_G(\lambda)$ y $p_H(\lambda)$ con respecto a de sus matrices asociadas, se dice que son coespectrales si $p_G(\lambda)=p_H(\lambda)$, es decir, comparten los mismos valores propios. Llamaremos familia coespectral al conjunto de dos o más gráficas que comparten el mismo polinomio característico. Si una familia solo contiene una gráfica, se denomina familia unitaria. Las familias coespectrales agrupan gráficas que, pese a no ser isomorfas, comparten el mismo polinomio característico y, en consecuencia, idéntico espectro [5]. Este fenómeno revela limitaciones en el poder de los invariantes espectrales para distinguir estructuras, al mismo tiempo que ofrece un marco para estudiar propiedades compartidas entre gráficas distintas [6].

3. DESCRIPCIÓN DEL ESTUDIO

En este trabajo se estudió el comportamiento espectral de todas las gráficas no isomorfas con hasta $n=10$ vértices. El propósito fue obtener y organizar las familias coespectrales correspondientes, con el fin de analizar su distribución, describir sus tamaños, identificar familias extremas y explorar las propiedades estructurales que se reflejan en el espectro.

Para cada gráfica se calculó el polinomio característico de la matriz de adyacencia y de la matriz laplaciana, con el fin de clasificar de manera exacta las familias. Además, se calcularon numéricamente los valores propios de cada familia a partir de una matriz representativa, con el propósito de estudiar las características estructurales compartidas en cada familia.

4. METODOLOGÍA

4.1. Métodos numéricos utilizados

En esta sección se describen los métodos numéricos utilizados para el cálculo del polinomio característico y de los valores propios de las matrices asociadas. Aunque ambos algoritmos son conocidos en la literatura, se presentan de manera resumida con el fin de contextualizar su aplicación en este trabajo y destacar las ventajas que ofrecen en el análisis espectral de gráficas.

Método de Samuelson – Berkowitz

La siguiente subsección es una adaptación del método presentado en [7].

El algoritmo de Samuelson–Berkowitz, constituye un procedimiento puramente algebraico para el cálculo de los coeficientes del polinomio característico de una matriz cuadrada. A diferencia de otros métodos clásicos, su principal ventaja es que no requiere divisiones ni la evaluación directa de determinantes, lo que lo hace aplicable en anillos conmutativos arbitrarios.

Sea $A \in [K]_{n\times n}$ una matriz con entradas en un anillo conmutativo K. El objetivo del algoritmo es calcular todos los coeficientes del polinomio característico:

$$p_A(\lambda)=\det(\lambda\cdot I_n-A)=\lambda^n+c_{n-1}\cdot\lambda^{n-1}+\cdots+c_1\cdot\lambda+c_0$$

de manera puramente algebraica, sin recurrir al cálculo directo de determinantes ni a operaciones con divisiones.

El método se apoya en una descomposición recursiva de la matriz A. Para cada nivel $k=0,\ldots,n-1$, se considera una submatriz A_k de dimensión $(n-k)\times(n-k)$, que se divide en bloques de la forma:

$$A_k=\begin{bmatrix} a_{11}^{(k)} & R^{(k)} \\ C^{(k)} & A_{k+1} \end{bmatrix}$$

Aquí $a_{11}^{(k)}$ es el primer elemento de la diagonal principal, $R^{(k)}$ es el resto de la primera fila, $C^{(k)}$ el resto de la primera columna y A_{k+1} es la submatriz obtenida al eliminar la primera fila y columna de A_k. El proceso comienza con $A_0=A$ y concluye cuando se alcanza una matriz de dimensión 1×1.

A partir de esta descomposición, en cada nivel k se construye una matriz rectangular:

$$T_k\in[K]_{(m+1)\times m},\ \text{si}\ \ A_k=[K]_{m\times m}$$

Con la siguiente estructura:

$$T_k=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -a_{11}^{(k)} & 1 & 0 & \cdots & 0 \\ -\alpha_0^{(k)} & -a_{11}^{(k)} & 1 & \cdots & 0 \\ -\alpha_1^{(k)} & -\alpha_0^{(k)} & -a_{11}^{(k)} & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ -\alpha_{m-2}^{(k)} & -\alpha_{m-3}^{(k)} & \cdots & -\alpha_0^{(k)} & -a_{11}^{(k)} \end{bmatrix}$$

La matriz T_k tiene estructura de tipo Toeplitz inferior: su diagonal principal contiene unos, la primera subdiagonal está compuesta por el valor $-a_{11}^{(k)}$, y las subdiagonales siguientes por los escalares $-\alpha_j^{(k)}$. Estos valores se definen como:

$$a_j^{(k)}=R^{(k)}\cdot A_{k+1}^{j}\cdot C^{(k)},\qquad j=0,1,\ldots,m-2$$

En general, $\alpha_j^{(k)}$ se obtiene como producto de $R^{(k)}$ con el resultado de aplicar A_{k+1} de manera iterada a $C^{(k)}$.

Para evitar la potenciación directa de la matriz A_{k+1} en cada $a_j^{(k)}$ se puede definir un vector u_j de manera iterativa de la siguiente forma:

$$u_0 = C^{(k)}$$
$$u_{j+1} = A_{k+1} \cdot u_j, \qquad j \geq 0$$

De este modo, se cumple que:

$$u_j = A_{k+1}^j \cdot C^{(k)}$$

Por lo tanto, los escalares $\alpha_j^{(k)}$ se reescriben como:

$$\alpha_j^{(k)} = R^{(k)} \cdot u_j, j = 0, 1, \ldots, m-2$$

Finalmente, el producto matricial:

$$v = T_0 \cdot T_1 \cdot \ldots \cdot T_{n-1} \in K^{n+1}$$

proporciona directamente el vector de coeficientes del polinomio característico de A. Más explícitamente:

$$v = \begin{bmatrix} 1 \\ c_{n-1} \\ c_{n-2} \\ \vdots \\ c_0 \end{bmatrix}$$

Este método ofrece una ventaja particular en el contexto de gráficas: dado que las matrices de adyacencia tienen como entradas únicamente ceros y unos, es posible trabajar directamente en el anillo de los enteros para calcular los coeficientes. Con ello se obtiene un cálculo exacto utilizando únicamente variables de tipo enteras, evitando la necesidad de recurrir a tipos de datos en punto flotante y a los problemas de precisión que estos conllevan.

Método de Jacobi para valores propios

La siguiente subsección es una adaptación del método presentado en [8].

El método de Jacobi fue formulado por Carl Gustav Jacob Jacobi en 1846 como un procedimiento iterativo para diagonalizar matrices simétricas reales mediante rotaciones ortogonales. En su época resultaba impráctico, pues cada paso requiere una gran cantidad de operaciones manuales, de modo que fue considerado más una curiosidad teórica que un algoritmo de cálculo efectivo. Sin embargo, a partir de la década de 1950, con la llegada de las primeras computadoras electrónicas, el método comenzó a ser empleado de manera sistemática. Su simplicidad algebraica, la conservación de la ortogonalidad y la estabilidad numérica de sus transformaciones lo convirtieron en una herramienta moderna y práctica para el cálculo de valores propios de matrices simétricas.

Sea $A \in R^{n \times n}$ una matriz simétrica real. El método construye una sucesión de matrices semejantes de la forma:

$$A^{(k+1)} = P^{(k)} A^{(k)} \left(P^{(k)}\right)^T, \qquad A^{(0)} = A$$

donde cada $P^{(k)}$ es una matriz ortogonal de rotación de Givens que actúa únicamente en un par de índices (p, q) con $p \neq q$. La idea es anular progresivamente los elementos fuera de la diagonal hasta aproximar $A^{(k)}$ a una matriz diagonal.

Para ello, en cada iteración se selecciona un par (p, q) y se define un ángulo θ tal que

$$\tan(2\theta) = \frac{2a_{pq}^{(k)}}{a_{pp}^{(k)} - a_{qq}^{(k)}}, \qquad c = \cos(\theta), \qquad s = \sin(\theta)$$

con lo cual la transformación garantiza que el nuevo elemento $a_{pq}^{(k+1)} = 0$. Los demás elementos de la matriz se actualizan de acuerdo con estas rotaciones, conservando la simetría en cada paso.

El proceso continúa hasta que todos los elementos fuera de la diagonal satisfacen $\left|a_{ij}^{(k)}\right| < \varepsilon$ para $i \neq j$, con una tolerancia ε fijada previamente. En ese punto, la matriz obtenida es prácticamente diagonal y sus entradas diagonales constituyen los valores propios de A. Los vectores propios se obtienen simultáneamente como las columnas del producto acumulado de todas las rotaciones:

$$V = P^{(0)} P^{(1)} \cdots P^{(k)}$$

Aunque la complejidad del método es cúbica en el tamaño de la matriz, su alta estabilidad y facilidad de implementación lo hacen particularmente útil en problemas donde las matrices son de tamaño moderado. En este trabajo, esa condición se cumple de manera natural, ya que las matrices de adyacencia y laplacianas de las gráficas consideradas no superan el orden de 11×11. De este modo, el método de Jacobi permitió calcular con precisión los valores propios representativos de cada familia coespectral, ofreciendo información confiable sobre sus propiedades estructurales.

4.2 Desarrollo del proyecto

Cálculo de gráficas no isomorfas

Para obtener todas las gráficas no isomorfas de n vértices se empleó el paquete Nauty and Traces, ejecutado desde una terminal de Ubuntu bajo el entorno WSL en Windows. Este software permite generar de forma eficiente todas las gráficas no isomorfas posibles para un número dado de vértices, garantizando que cada representación aparezca una sola vez [9].

Las gráficas se almacenaron en el formato graph6, un estándar ampliamente utilizado por su compacidad y facilidad para ser interpretado por distintos programas de análisis gráfico. Para mantener una estructura organizada, se generó un archivo independiente por cada valor de n, con extensión ".g6".

La generación se realizó mediante la instrucción:

```
geng  n  > n.g6
```

para $1 \leq n \leq 11$.

El uso de Nauty garantiza la generación completa y sin repeticiones de todas las gráficas no isomorfas de orden n, mediante algoritmos certificados de canonización e isomorfismo.

En el caso particular de $n=11$, el proceso produjo un archivo de aproximadamente 11.3GB, que contiene 1,018,997,864 gráficas no isomorfas. Este volumen de datos refleja el crecimiento exponencial del número de gráficas con respecto a n, y justifica el uso posterior de técnicas de programación concurrente para el manejo y procesamiento eficiente de la información.

Arquitectura del proyecto y programación concurrente

El programa fue desarrollado en C++23 utilizando únicamente la biblioteca estándar, con el propósito de mantener portabilidad, eficiencia y control directo sobre la administración de memoria. Su diseño siguió un enfoque modular, donde cada componente opera de forma independiente para facilitar la depuración y la extensión del código a nuevos casos. La estructura general comprende las etapas de lectura de archivos, decodificación de gráficas, procesamiento concurrente cuando es necesario, cálculo del polinomio característico y clasificación final de las familias coespectrales.

El procesamiento se realiza de manera iterativa para cada conjunto de gráficas no isomorfas generado por Nauty. En total, se ejecutaron diez iteraciones correspondientes a $1 \leq n \leq 10$. En cada una se lee el archivo n.g6, se decodifican las cadenas en formato graph6 y se transforman en sus respectivas matrices de adyacencia mediante un procedimiento que reconstruye la información binaria del formato comprimido. Las matrices se representan de forma compacta mediante vectores de bits, lo que reduce considerablemente el uso de memoria frente a representaciones convencionales y optimiza la velocidad de acceso.

Todas las matrices se almacenan en un contenedor contiguo en memoria, evitando accesos frecuentes al disco. Este enfoque es viable hasta $n=10$, donde el número total de gráficas no isomorfas puede mantenerse íntegramente en memoria principal. En la

práctica, esta decisión permitió aprovechar la localidad de referencia del procesador y mejorar notablemente el rendimiento.

El cálculo de los polinomios característicos se basa en el método de Samuelson–Berkowitz descrito anteriormente. Los resultados obtenidos se encapsulan en estructuras que asocian los coeficientes del polinomio con el identificador correspondiente de cada matriz, formando un registro completo para cada gráfica procesada.

El programa ajusta automáticamente su modo de ejecución en función del tamaño del conjunto de gráficas. Para los casos pequeños ($n \leq 8$) se emplea ejecución secuencial, mientras que para los casos de mayor tamaño se activa un esquema concurrente. En este último, las matrices se dividen en bloques que se distribuyen entre los núcleos disponibles del procesador. Cada hilo procesa su bloque de manera independiente y sincroniza las operaciones de escritura mediante mecanismos de exclusión mutua, lo que asegura la coherencia de los datos y evita condiciones de carrera.

La ejecución concurrente permitió reducir significativamente los tiempos de cómputo en los casos más costosos, manteniendo la precisión de los resultados y garantizando escalabilidad en entornos multinúcleo. En particular, para n = 10 el tiempo de ejecución se redujo aproximadamente en un factor cercano al número de núcleos físicos disponibles, pasando de un tiempo estimado superior a 30 minutos en ejecución secuencial a 324 segundos en ejecución concurrente. Una vez finalizado el cálculo, los registros se ordenan por sus coeficientes para agrupar las gráficas que comparten el mismo polinomio característico, conformando así las familias coespectrales. Los resultados se exportan a archivos organizados por número de vértices y acompañados de métricas de tiempo y consumo de memoria obtenidas durante la ejecución.

Aunque sería posible optimizar la clasificación final mediante estructuras de dispersión, se optó por conservar el esquema de ordenamiento secuencial para mantener la generalidad del código, especialmente ante el incremento de tamaño en el caso $n=11$. De esta forma, el núcleo del programa se mantuvo inalterado y sirvió como base directa para las extensiones posteriores.

La arquitectura general del proyecto se muestra en la Figura 1, que resume visualmente la estructura modular y las etapas de procesamiento concurrente descritas en esta sección.

De manera análoga, para el caso de las matrices laplacianas se aplicó el mismo esquema general. La diferencia se encuentra en la etapa de cálculo, donde el programa construye previamente la matriz laplaciana a partir de la relación

$$L = D - A$$

siendo D la matriz diagonal de grados y A la matriz de adyacencia. Una vez generada, la matriz laplaciana se procesa mediante el mismo método algebraico, siguiendo un flujo idéntico al de las matrices de adyacencia, con la única diferencia de que los resultados se almacenan en archivos separados para mantener la independencia entre ambos tipos de espectro.

Caso de familias de $n=11$ vértices

El caso de $n=11$ representa un incremento sustancial en complejidad, con 1,018,997,864 gráficas no isomorfas. En este punto, tanto la carga completa en memoria como el almacenamiento simultáneo de los polinomios característicos resultan inviables. Para resolverlo, se mantuvo la misma arquitectura del programa, pero adaptada a un esquema híbrido que combina concurrencia con acceso directo a disco.

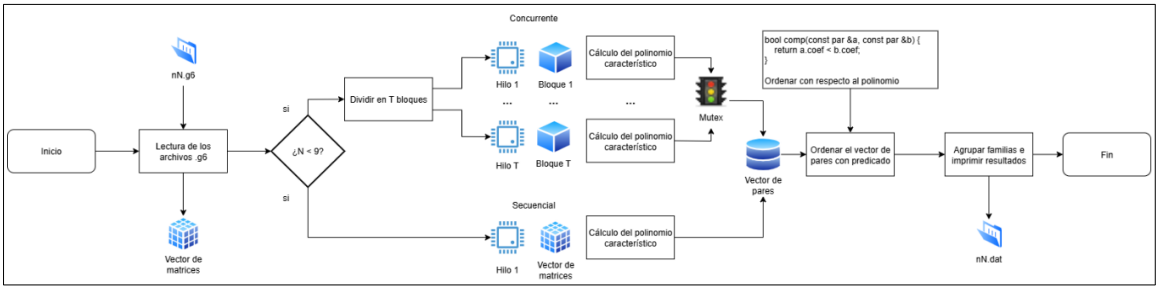

Figura 1. Esquema visual de la arquitectura del programa. Representa la lectura de las gráficas, el cálculo de los polinomios característicos y la construcción concurrente de las familias coespectrales.

El archivo 11.g6 contiene todas las gráficas codificadas en formato graph6. Cada hilo procesa una sección del archivo definida por bloques de tamaño fijo, calculado a partir de la memoria disponible mediante la expresión

$$M_T = \frac{M_h}{T(13I + 12C)}$$

donde M_h es la memoria total en bytes, T el número de hilos, I el tamaño de un entero y C el de un carácter. Esta relación permite determinar cuántas matrices puede manejar cada hilo sin superar su límite de memoria asignado sabiendo que cada matriz está codificada por 11 caracteres más el salto de línea y cada resultado serán 12 enteros del coeficiente del polinomio más el entero que representa el id de la matriz.

El acceso concurrente al archivo se gestiona mediante un contador global de posición protegido por exclusión mutua. Cada hilo obtiene su rango de trabajo a partir de este contador, registrando una copia local de la posición inicial desde la cual realiza las operaciones de lectura y escritura. Con esta información, cada hilo lee directamente su bloque, lo decodifica y realiza el cálculo del polinomio característico, almacenando los coeficientes y el identificador en una estructura binaria compacta.

Los resultados de cada bloque se escriben en posiciones específicas del archivo intermedio pre11adj.dat, utilizando desplazamientos controlados que garantizan que cada hilo trabaje en una región independiente del archivo. Este procedimiento asegura que la escritura sea concurrente y consistente, sin interferencias entre hilos.

Una vez completado el procesamiento de todas las gráficas, el archivo intermedio se somete a un ordenamiento externo mediante mezcla de k-bloques, con ordenamiento concurrente dentro de cada bloque y mezcla secuencial al final. El archivo resultante, adj11.dat, contiene los polinomios característicos y los identificadores ordenados, lo que permite un recorrido final para agrupar las gráficas en sus respectivas familias coespectrales.

Esta versión conserva la misma estructura modular de las anteriores, trasladando la gestión de datos desde la memoria principal hacia el disco sin alterar la lógica general del sistema. El uso de bloques independientes, acceso directo y sincronización mínima permitió procesar volúmenes de datos del orden de mil millones de elementos sin reescribir la arquitectura del programa.

4.3. Condiciones experimentales

Todas las pruebas se realizaron en un equipo con procesador Intel® Core™ i7-8700, de seis núcleos y doce hilos lógicos, con frecuencia máxima de 4.6 GHz y 12 MB de memoria caché [10]. El sistema contaba con 32 GB de memoria RAM DDR4 y una unidad

de estado sólido (SSD) de 512 GB para almacenamiento. El entorno de ejecución correspondió a Windows 11 (64 bits), y el código fue compilado con GCC 14.2.0 [11], empleando el estándar C++23.

Las mediciones de tiempo y consumo de memoria reportadas se obtuvieron mediante cronómetros internos del programa y el monitoreo del uso de recursos del sistema operativo.

5. RESULTADOS

5.1. Resultados del espectro de adyacencia

El análisis global muestra que las familias coespectrales de adyacencia comienzan a aparecer a partir de $n=5$, y su número crece de manera constante con el orden de la gráfica.

Como se aprecia en la Tabla 1, la proporción de gráficas que pertenecen a alguna familia coespectral pasa de 0.05 para $n=5$ a 0.21 para $n=10$, lo que indica que la coincidencia espectral se vuelve cada vez más frecuente conforme aumenta la complejidad estructural del conjunto.

Tabla 1. Proporción global de familias coespectrales de adyacencia.

N	Total de gráficas	Familias coespectrales	Familias unitarias	Proporción Coespectrales /Total
1	1	0	1	0
2	2	0	2	0
3	3	0	3	0
4	11	0	11	0
5	34	1	32	0.05
6	156	5	146	0.06
7	1044	54	934	0.10
8	12346	829	10624	0.13
9	274668	23728	223629	0.18
10	12005168	1163566	9444562	0.21

En cuanto a la estructura interna de las familias, los resultados de la Tabla 2 muestran que la mayoría de ellas son muy pequeñas, con un tamaño promedio cercano a 2 en todos los casos. La dispersión crece con el orden, lo que permite la aparición de familias más extensas: para $n=9$ el tamaño máximo alcanza 10 gráficas, y para $n=10$ llega hasta 21, que corresponde a la familia más grande identificada en todo el estudio. En conjunto, los datos indican que, aunque la coespectralidad se vuelve más común con n, la mayoría de las gráficas coespectrales siguen agrupándose en pares o tríos, y solo en órdenes altos aparecen familias de tamaño considerable.

Tabla 2. Tamaño promedio y máximos de las familias coespectrales de adyacencia.

N	Tamaño promedio de familia	Desviación estándar del tamaño de las familias	Tamaño máximo de familia
5	2.0	0.0	2
6	2.0	0.0	2
7	2.04	0.19	3
8	2.08	0.29	4
9	2.15	0.47	10
10	2.20	0.60	21

Las estadísticas globales del espectro de adyacencia, resumidas en la Tabla 3, presentan un comportamiento coherente con las propiedades teóricas. La media de los valores propios se mantiene próxima a cero para todos los órdenes, lo que refleja la traza nula de la matriz de adyacencia. La desviación estándar aumenta desde 0.71 ($n=2$) hasta 2.12 ($n=10$), mostrando un ensanchamiento progresivo del espectro. La asimetría es positiva y crece con el orden (de 0.5 a 0.96), lo que revela una distribución más cargada hacia valores positivos, característica del predominio de gráficas no bipartitas. La curtosis, por su parte, tiende a estabilizarse alrededor de $3-3.6$, lo que indica que la forma global del espectro se aproxima a una distribución normal. Los valores mínimo y máximo respetan las cotas teóricas $\pm(n-1)$, y la energía promedio normalizada se estabiliza cerca de 1.63 para n grandes, mientras que la energía total crece casi linealmente con n.

La Figura 2 muestra de manera visual la evolución de estas distribuciones. Las curvas representan la frecuencia relativa de los valores propios en intervalos de una unidad y evidencian una alta concentración alrededor de $\lambda=0$. A medida que aumenta n, la

concentración central disminuye y las colas se extienden hacia los extremos $\pm(n-1)$, lo que refleja una dispersión espectral creciente. La disminución en la altura de los picos y el ensanchamiento de las curvas son consistentes con el aplanamiento de la distribución observado en la Tabla 3.

Para ilustrar casos específicos, la Figura 3 muestra la familia coespectral más pequeña calculada, correspondiente a n=5, formada por dos gráficas distintas que comparten el polinomio característico $\lambda^5-4\lambda^3$. El espectro asociado a este polinomio es $\{-2,0,0,0,2\}$, que al ser simétrico con respecto al cero significa que son bipartitas, y ambas gráficas presentan estructuras diferentes pero idéntico conjunto de valores propios.

Tabla 3. Estadísticas globales del espectro de adyacencia.

N	Media	Desv. est.	Asimetría	Curtosis	Mínimo	Máximo	Energía prom. normalizada	Energía promedio
1	0	0	NaN	NaN	0	0	0	0
2	0.00	0.71	0.00	1.50	-1.00	1.00	0.50	1.00
3	0.00	1.00	0.50	-0.33	-1.41	2.00	0.74	2.21
4	0.00	1.22	0.59	-0.29	-2.00	3.00	0.93	3.71
5	0.00	1.42	0.73	-0.12	-2.45	4.00	1.09	5.43
6	0.00	1.59	0.77	-0.01	-3.00	5.00	1.22	7.32
7	0.00	1.74	0.82	0.12	-3.46	6.00	1.34	9.38
8	0.00	1.88	0.86	0.26	-4.00	7.00	1.45	11.58
9	0.00	2.00	0.91	3.43	-4.47	8.00	1.54	13.89
10	0.00	2.12	0.96	3.62	-5.00	9.00	1.63	16.31

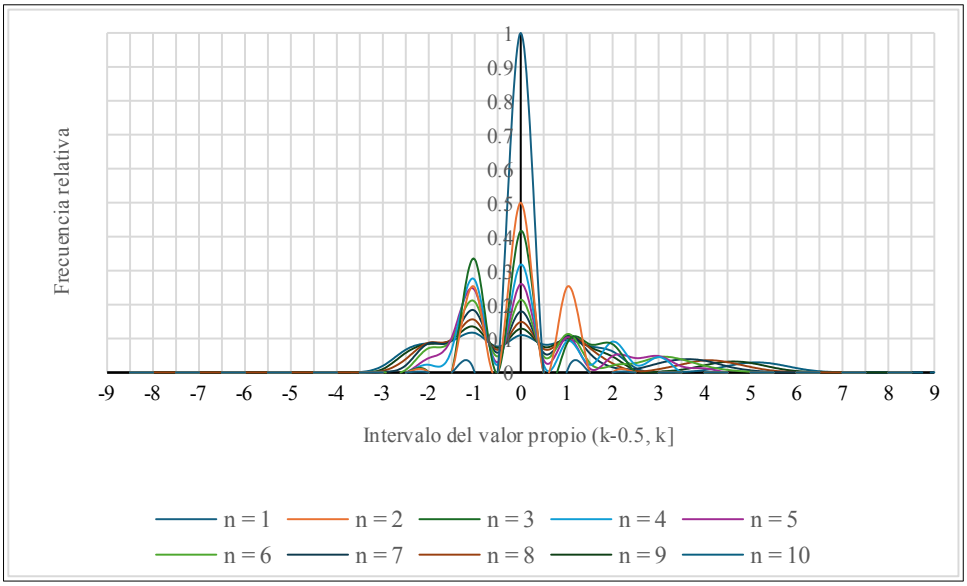

Figura 2. Distribución relativa del espectro de adyacencia.

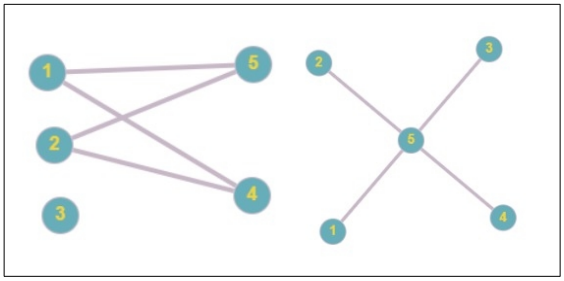

Figura 3. Familia coespectral de adyacencia más pequeña.

Por otro lado, la Figura 4 muestra una de las dos familias coespectrales más grandes calculadas, con n = 10 y 21 gráficas distintas. El polinomio característico común es

$$\lambda^{10}-20\lambda^8-18\lambda^7+84\lambda^6+76\lambda^5-119\lambda^4-72\lambda^3+56\lambda^2,$$

y los valores propios resultantes son aproximadamente

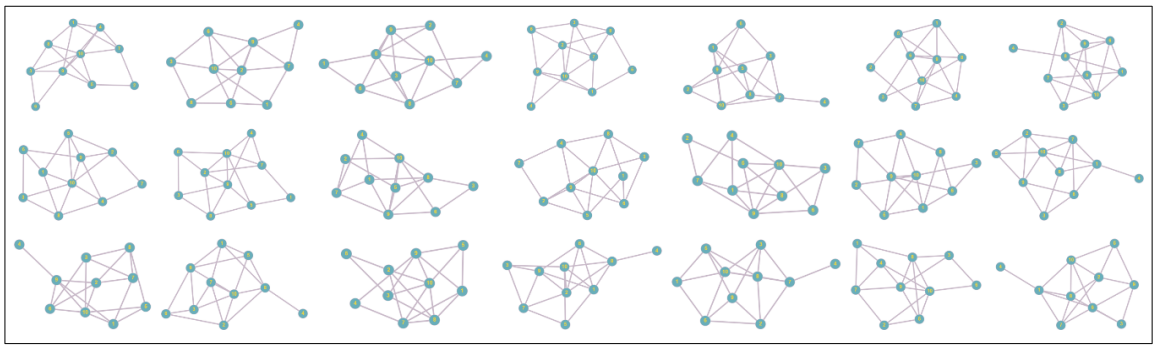

Figura 4. Una de las dos familias coespectrales de adyacencia más grande calculada.

$\{-2.66, -2.29, -1.53, -1.30, 0, 0, 0.54, 1.16, 1.69, 4.38\}$

En este caso, el espectro presenta valores cercanos a cero, pero no es simétrico respecto al origen, lo que indica que las gráficas de esta familia no son bipartitas.

En conjunto, los resultados de adyacencia muestran que la coespectralidad crece con el orden, que la mayoría de las familias son pequeñas y que el espectro global se ensancha y se aproxima a una forma normal conforme n aumenta.

5.2. Resultados del espectro laplaciano

El análisis laplaciano presenta un comportamiento análogo en tendencia general, aunque con diferencias cuantitativas importantes. De acuerdo con la Tabla 4, la proporción de gráficas que comparten el mismo espectro laplaciano es menor que en el caso de adyacencia, situándose entre 0.05 y 0.07 para los órdenes mayores. Esto sugiere que el espectro laplaciano distingue con mayor precisión entre estructuras no isomorfas, reduciendo la probabilidad de coincidencia espectral.

En la Tabla 5 se observa que el tamaño promedio de las familias se mantiene alrededor de 2.1 para $n \geq 7$, con desviaciones estándar similares a las obtenidas en adyacencia. Sin embargo, el tamaño máximo llega a 23 gráficas para $n=10$, lo que demuestra que también en el caso laplaciano existen familias extensas, aunque son poco frecuentes.

Tabla 4. Proporción global de familias coespectrales laplacianas

N	Total de gráficas	Familias coespectrales	Familias unitarias	Proporción Coespectrales/ Total
1	1	0	1	0
2	2	0	2	0
3	3	0	3	0
4	11	0	11	0
5	34	0	34	0
6	156	2	152	0.01
7	1044	62	914	0.06
8	12346	819	10579	0.07
9	274668	19729	232073	0.07
10	12005168	656784	11249514	0.05

Tabla 5. Tamaño promedio y máximos de las familias coespectrales laplacianas.

N	Tamaño promedio de familia	Desviación estándar del tamaño de las familias	Tamaño máximo de familia
6	2	0	2
7	2.09	0.29	3
8	2.15	0.46	6
9	2.15	0.49	9
10	2.15	0.54	23

Las estadísticas del espectro laplaciano, presentadas en la Tabla 6, cumplen las propiedades esperadas: el mínimo es siempre 0, asociado al valor propio trivial; la media crece linealmente con n, alcanzando 4.5 en $n=10$; y la desviación estándar aumenta de manera sostenida hasta 2.62, reflejando el ensanchamiento del rango espectral. La asimetría se mantiene cercana a cero y tiende a valores ligeramente negativos en los órdenes altos, mientras que la curtosis se mantiene moderada $(\approx 2-3)$, lo que indica una distribución más equilibrada y menos sesgada que en la adyacencia.

La Figura 5 ilustra la distribución relativa de los valores propios laplacianos para los distintos órdenes. A diferencia del caso anterior, las curvas presentan picos dominantes en valores bajos, con una fuerte acumulación cerca de $\lambda=0$ y una dispersión gradual hacia la derecha. Esta característica refleja la naturaleza no negativa del espectro laplaciano y su tendencia a concentrar la mayor parte de los valores en el rango inferior. En comparación con la Figura 2, el espectro laplaciano es más asimétrico hacia valores pequeños y carece de simetría central, confirmando que ambos espectros describen comportamientos estructurales distintos.

Tabla 6. Estadísticas globales del espectro laplaciano.

N	Media	Desv. est.	Asimetría	Curtosis	Mínimo	Máximo	Energía prom. normalizada	Energía promedio
t	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.50	0.87	1.15	1.28	0.00	2.00	0.50	1.00
3	1.00	1.05	0.89	1.45	0.00	3.00	1.00	3.00
4	1.50	1.57	0.47	1.94	0.00	4.00	1.50	6.00
5	2.00	1.83	0.33	2.08	0.00	5.00	2.00	10.00
6	2.50	2.02	0.21	2.27	0.00	6.00	2.50	15.00
7	3.00	2.20	0.12	2.38	0.00	7.00	3.00	21.00
8	3.50	2.35	0.03	2.45	0.00	8.00	3.50	28.00
9	4.00	2.48	-0.05	2.54	0.00	9.00	4.00	36.00
10	4.50	2.62	-0.12	2.66	0.00	10.00	4.50	45.00

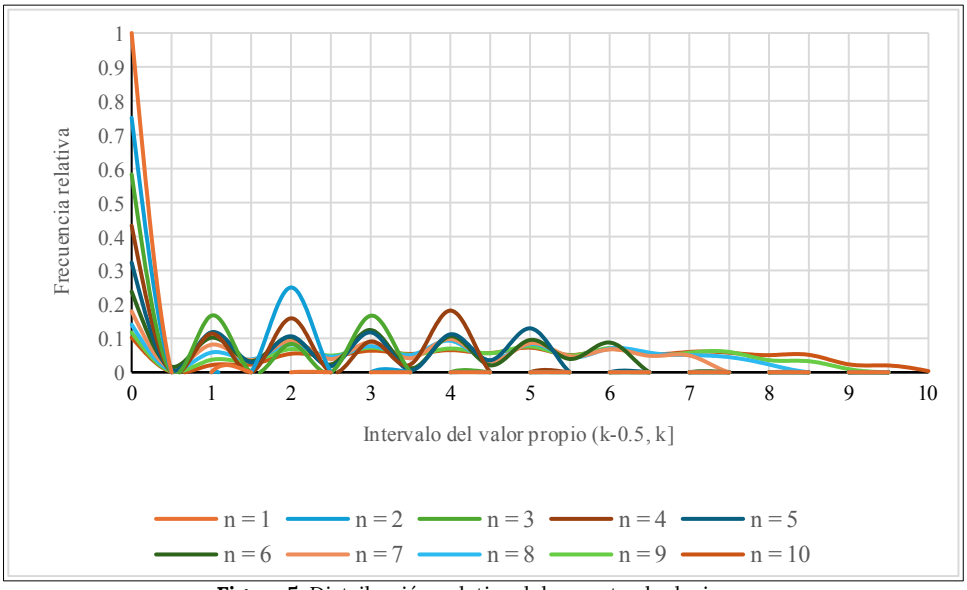

Figura 5. Distribución relativa del espectro laplaciano.

En conjunto, las Tablas 1–6 y las Figuras 2–5 evidencian que la coespectralidad es un fenómeno poco frecuente pero creciente, con familias que se mantienen pequeñas en promedio y distribuciones espectrales que se ensanchan progresivamente con el orden de la gráfica. El espectro de adyacencia se caracteriza por su simetría aproximada y su asimetría positiva, mientras que el espectro laplaciano concentra la masa en valores bajos y muestra una mayor estabilidad.

Para los resultados solo se analizaron las familias coespectrales de hasta $n=10$ vértices, puesto que el caso $n=11$ requería de más memoria en disco de la que tenía el hardware. El tiempo de ejecución y el uso de memoria para el cálculo de las familias coespectrales (para cada tipo de matriz) se pueden ver en la siguiente tabla:

Tabla 7. Uso de tiempo y memoria para el cálculo de las familias coespectrales

Tamaño	Uso de hilos	Tiempo	Memoria
1	No	<0.0001s	<0.00001MB
2	No	<0.0001s	<0.00001MB
3	No	<0.0001s	0.0001MB
4	No	<0.0001s	0.0004MB
5	No	<0.0001s	0.001MB
6	No	0.009s	0.007MB
7	No	0.049s	0.051MB
8	Si	0.885s	0.659MB
9	Si	5.264s	15.716MB
10	Si	324.859s	732.737MB
11	Si	19 hrs	20GB en RAM + 250GB en disco

6. DISCUSIÓN

Los resultados obtenidos muestran con claridad cómo la coespectralidad y la estructura de los espectros evolucionan con el tamaño de las gráficas, pero también reflejan el esfuerzo técnico necesario para hacer posible su cálculo completo. Desde el punto de vista teórico, la aparición de familias coespectrales a partir de $n=5$ y su crecimiento con el orden evidencian el aumento de la complejidad combinatoria del espacio de posibles conexiones. Aunque la proporción de gráficas coespectrales crece con n, la mayoría de las gráficas siguen siendo espectralmente únicas, lo que reafirma el carácter fuertemente distintivo del espectro como invariante estructural.

El tamaño promedio de las familias se mantiene estable, cercano a dos elementos, lo que muestra que la coespectralidad tiende a presentarse de manera localizada, sin formar agrupamientos masivos. Aun así, en los órdenes más altos aparecen familias extensas, como la de 21 gráficas en el caso de la adyacencia, y la de 23 en el laplaciano. Estas familias grandes son excepcionales, pero confirman que es posible que distintas configuraciones estructurales compartan un mismo conjunto de valores propios.

Desde el punto de vista estadístico, las propiedades del espectro de adyacencia concuerdan con la teoría espectral de gráficas. La media nula y la desviación estándar creciente indican un ensanchamiento progresivo del espectro conforme aumenta n, mientras que la asimetría positiva y la curtosis cercana a tres muestran que la distribución de valores propios tiende a una forma casi normal. En contraste, el espectro laplaciano presenta un comportamiento más regular y estable: las energías se distribuyen de forma uniforme en el rango espectral, las curvas son suaves y equilibradas, y la concentración dominante en valores pequeños de λ se mantiene consistente a lo largo de los órdenes. Esta regularidad genera una estructura visual armónica, en la que la energía se reparte de manera ordenada y sin picos abruptos.

El desarrollo computacional del proyecto fue igualmente relevante. Hasta $n=10$, el análisis completo fue posible utilizando técnicas concurrentes y procesamiento en memoria, pero el paso hacia $n=11$ implicó un cambio de escala radical. El número de gráficas no isomorfas supera los mil millones, lo que impide realizar un procesamiento exhaustivo en una sola sesión o mantener los datos íntegramente en memoria. Ante ello, se diseñó una nueva arquitectura de ejecución basada en procesamiento por bloques, acceso directo a disco, sincronización mediante exclusión mutua y manejo de tareas concurrentes controladas. La corrección del procedimiento para $n=11$ se garantiza porque el algoritmo conserva exactamente la misma lógica algebraica utilizada en los casos $n\leq 10$, variando únicamente el esquema de almacenamiento y acceso a datos. Además, el ordenamiento externo mantiene la agrupación exacta de polinomios característicos, por lo que la clasificación no depende del tamaño del conjunto sino únicamente de la igualdad algebraica de coeficientes.

En conjunto, los resultados teóricos y computacionales muestran la coherencia entre la evolución de las propiedades espectrales y el crecimiento de la complejidad combinatoria, y al mismo tiempo evidencian

que el diseño de herramientas concurrentes es esencial para extender los límites prácticos del análisis coespectral. La arquitectura implementada constituye así un punto de partida sólido para futuras investigaciones que deseen retomar el análisis completo de los datos generados para $n=11$ o escalar el estudio hacia órdenes mayores.

7. CONCLUSIONES

El análisis realizado confirma que el espectro, aunque contiene información estructural relevante, no determina de manera única a la gráfica. Existen casos en los que dos gráficas con estructuras muy distintas comparten el mismo conjunto de valores propios, lo que demuestra que el espectro por sí solo no es suficiente para distinguirlas. Este hecho justifica el interés en estudiar las familias coespectrales: conjuntos de gráficas no isomorfas que, pese a sus diferencias, presentan idéntico comportamiento espectral.

El trabajo permitió calcular y analizar exhaustivamente todas las familias coespectrales de gráficas no isomorfas hasta 10 vértices, integrando métodos numéricos con programación concurrente. Los resultados muestran que la coespectralidad aparece a partir de $n=5$ y aumenta con el orden, aunque la mayoría de las gráficas permanecen espectralmente únicas. La mayoría de las familias son pequeñas, con un tamaño promedio constante, y solo en los órdenes altos se identifican agrupamientos amplios. El espectro de adyacencia conserva una media nula y presenta una desviación estándar creciente, reflejando la expansión progresiva del espectro y la tendencia hacia una distribución normal. En contraste, el espectro laplaciano muestra una organización más equilibrada, con energías distribuidas uniformemente y una estabilidad que refuerza su carácter como descriptor estructural más regular.

Desde el punto de vista computacional, el desarrollo del sistema confirmó la importancia de la concurrencia para el análisis de grandes volúmenes de gráficas. La implementación fue capaz de procesar de manera eficiente los conjuntos completos hasta $n=10$ y dejó establecida una arquitectura funcional para $n=11$, que permite generar y agrupar las familias coespectrales correspondientes. Si bien estos datos no fueron analizados en esta etapa, la estructura del programa y su capacidad de procesamiento quedan disponibles para su uso futuro. En ese sentido, la implementación constituye una herramienta abierta que puede ser utilizada y extendida por otros investigadores para continuar el análisis estadístico y estructural de $n=11$ o explorar órdenes mayores sin modificar la base del sistema.

En conjunto, el estudio consolida una visión completa del fenómeno de la coespectralidad y demuestra que el equilibrio entre modelación algebraica, análisis estadístico y diseño concurrente constituye un enfoque eficaz para explorar espacios combinatorios de gran escala. La metodología desarrollada y la arquitectura implementada sientan las bases para ampliar el análisis hacia órdenes superiores y para estudiar otras matrices asociadas, fortaleciendo la relación entre la teoría espectral de gráficas y el cómputo científico contemporáneo.

8. APÉNDICE

Los programas utilizados y la información de las gráficas de $n \leq 8$ vértices se encuentran disponibles en el repositorio general de GitHub para este trabajo:

https://github.com/danielramosl/FamiliasCoespectrales.

REFERENCIAS

[1] Chacón JL. Introducción a la teoría de grafos. Granada: Universidad de Granada; 2005 [citado 2025 Nov 1].

https://www.studocu.com/es-mx/document/universidad-tecnologica-de-ciudad-juarez/bases-de-datos-para-aplicaciones/introduccion-a-la-teoria-de-grafos-chacon-jose-luis/95938656
[2] Chung FRK. Spectral graph theory. Providence (RI): American Mathematical Society; 1997. doi: 10.1090/cbms/092
[3] Cvetković DM, Doob M, Sachs H. Spectra of graphs: theory and applications. 3rd ed. Berlin: Springer-Verlag; 1995. doi: 10.1007/978-1-4614-1939-6
[4] Jiang J. An introduction to spectral graph theory. Chicago (IL): University of Chicago; 2012 [citado 2025 Nov 1]. https://math.uchicago.edu/~may/REU2012/REUPapers/JiangJ.pdf
[5] Haemers WH, Spence E. Enumeration of cospectral graphs. European Journal of Combinatorics (2004) 25(2):199-211. doi: 10.1016/S0195-6698(03)00100-8
[6] Van Dam ER, Haemers WH. Which graphs are determined by their spectrum? Linear Algebra and its Applications (2003) 373:241-272. doi: 10.1016/S0024-3795(03)00483-X
[7] Berkowitz SJ. On computing the determinant in small parallel time using a small number of processors. Information Processing Letters (1984) 18(3):147-150. doi: 10.1016/0020-0190(84)90018-8
[8] Rutishauser H. The Jacobi method for real symmetric matrices. En: Bauer FL, editor. Linear algebra. Handbook for Automatic Computation. Vol. 2. Berlin: Springer; 1971. p. 202-211. doi: 10.1007/978-3-662-39778-7_12
[9] McKay BD, Piperno A. Practical graph isomorphism II. Journal of Symbolic Computation (2014) 60:94-112. doi: 10.1016/j.jsc.2013.09.003
[10] Intel Corporation. Intel® Core™ i7-8700 Processor (12M Cache, up to 4.60 GHz) – Product Specifications. Intel; c2017 [citado 2025 Nov 1]. https://www.intel.la/content/www/xl/es/products/sku/126686/intel-core-i78700-processor-12m-cache-up-to-4-60-ghz/specifications.html
[11] Free Software Foundation; GCC Developer Community. Using the GNU Compiler Collection (for GCC version 14.2.0). Massachusetts: GNU Press/FSF; 2024 [citado 2025 Nov 1]. https://gcc.gnu.org/onlinedocs/

ACERCA DEL AUTOR

Luis Daniel Ramos López es estudiante de la Maestría en Ciencias Matemáticas Aplicadas e Industriales en la Universidad Autónoma Metropolitana Unidad Iztapalapa. Es licenciado en Ingeniería en Computación por la Universidad Autónoma Metropolitana Unidad Azcapotzalco, donde también se desarrolló como ayudante del Departamento de Sistemas. Sus intereses de investigación abarcan la programación concurrente y el cómputo de alto desempeño, el análisis y diseño de métodos numéricos, así como sus aplicaciones en la modelación matemática y el cómputo científico.