

**ELEMENTOS DE LA TEORÍA DE GRAFOS Y LA CONJETURA  
DE EVASIVIDAD**

**BIRON ENRIQUE ACOSTA CARVAJAL**

Código 0702-50192011

**LUISA MARÍA MONTOYA CONDE**

Código 0702-00202012

**UNIVERSIDAD DEL TOLIMA  
FACULTAD DE CIENCIAS BÁSICAS  
PROGRAMA DE MATEMÁTICAS CON ÉNFASIS EN ESTADÍSTICA  
IBAGUÉ  
2018**

**ELEMENTOS DE LA TEORÍA DE GRAFOS Y LA CONJETURA  
DE EVASIVIDAD**

**BIRON ENRIQUE ACOSTA CARVAJAL**

Código 0702-50192011

**LUISA MARÍA MONTOYA CONDE**

Código 0702-00202012

Trabajo de Grado para optar por el título de  
Profesional en Matemáticas con énfasis en Estadística

Director

**JUAN PABLO YÁÑEZ PUENTES**

Profesor del Departamento de Matemáticas y Estadística

**UNIVERSIDAD DEL TOLIMA  
FACULTAD DE CIENCIAS BÁSICAS  
PROGRAMA DE MATEMÁTICAS CON ÉNFASIS EN ESTADÍSTICA  
IBAGUÉ  
2018**



UNIVERSIDAD DEL TOLIMA

FACULTAD DE CIENCIAS

PROGRAMA DE MATEMÁTICAS CON ÉNFASIS EN ESTADÍSTICA

ACTA DE SUSTENTACIÓN TRABAJO DE GRADO

TÍTULO: ELEMENTOS DE LA TEORÍA DE GRAFOS Y CONJETURA DE EVASIVIDAD

AUTORES:

BIRON ENRIQUE ACOSTA CARVAJAL (Código: 070250192011)

LUISA MARÍA MONTOYA CONDE (Código: 070200202012)

DIRECTOR: JUAN PABLO YAÑEZ PUENTES

JURADOS: HECTOR GRANADA DÍAZ  
JESÚS ANTONIO ÁVILA

CALIFICACIÓN: 4.7 (CUATRO SIETE) MERITORIO

APROBÓ

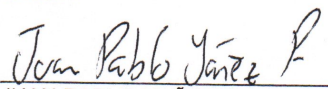
REPROBÓ

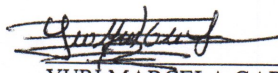
OBSERVACIONES:

FIRMAS:

  
HECTOR GRANADA DÍAZ  
Jurado 1

  
JESÚS ANTONIO ÁVILA  
Jurado 2

  
JUAN PABLO YAÑEZ PUENTES  
Director del Trabajo

  
YURI MARCELA GARCÍA S.  
Directora del Programa

Ciudad y fecha: Ibagué, 7 de mayo de 2018

6. CALIFICACIÓN

PRIMER JURADO:

NOMBRE DEL JURADO: HECTOR GRANADA DÍAZ

NOTA OTORGADA POR EL JURADO Cuatro siete (4,7)

FIRMA DEL JURADO [Firma]

SEGUNDO JURADO:

NOMBRE DEL JURADO: JESUS ANTONIO AVILA

NOTA OTORGADA POR EL JURADO Cuatro siete (4.7)

FIRMA DEL JURADO Jesús Antonio Avila G.

PROMEDIO FINAL DE LA NOTA DEL TRABAJO DE GRADO: Cuatro siete (4,7)

7. RANGOS DE EQUIVALENCIA: (Acuerdo No. 030 de 2000 del Consejo de Facultad)

Calificación menor de tres cero (3.0)

REPROBADO

Calificación entre tres cero (3.0) y tres nueve (3.9)

APROBADO

Calificación entre cuatro cero (4.0) y cuatro cuatro (4.4.)

SOBRESALIENTE

Calificación entre cuatro cinco (4.5) y cuatro nueve (4.9)

MERITORIO

Calificación de cinco cero (5.0)

LAUREADO

FECHA DE SUSTENTACIÓN

Mayo 7/2018

# Agradecimientos

Quiero agradecer a Dios quien me ha brindado salud y vida, también a mis padres Joaquín y Nelly, a mis hermanos Isabel y Juan Pablo quienes me han fortalecido sentimentalmente, a mis compañeros Luisa Madrigal quien me ha enseñado el valor de la humildad y la sencillez, Luisa Niño quien ha aportado sus conocimientos y paciencia, Biron Acosta por ser mi compañero de batalla durante este trabajo, Brayan Rubio quien es mi amigo fiel y juntos luchamos por Topología, William Chavarro por brindarme su apoyo moral y económico, por último al profesor Leonardo Solanilla por estar a mi lado en el momento que más lo necesité y por su hermosa amistad.

Luisa María Montoya Conde.

Quiero agradecer a Dios por darme la oportunidad de finalizar mis estudios, así como el conocimiento adquirido y la fortaleza para superar los obstáculos que se han presentado en mi vida. También quiero agradecer a mi madre Olga Liliana, por el apoyo que me brinda, a mi hermano Jarib Ricardo y mi hermana Eilin Sofía por su cariño incondicional. Agradezco a mi novia Luisa Madrigal por sus consejos y ánimos para no rendirme y seguir adelante en mis metas, a Luisa María por consejos y ser mi compañera de trabajo. A Luisa Niño por su apoyo y los debates matemáticos que tuvimos en toda la carrera y a Brayan Rubio por su amistad incondicional.

Biron Enrique Acosta Carvajal

En conjunto, agradecemos a las demás persona que han estado en el transcurso de nuestra carrera profesional. Y un agradecimiento especial a nuestro director Juan Pablo Yáñez por brindarnos sus conocimientos, tiempo y paciencia.

*Aprendí que el coraje no es la ausencia de miedo, sino el triunfo sobre él.  
El hombre valiente no es aquel que no siente miedo, sino el que conquista ese miedo.*

*Nelson Mandela*

# Índice general

<b>Agradecimientos</b>	<b>I</b>
<b>Introducción</b>	<b>1</b>
<b>1. Elementos de la teoría de grafos</b>	<b>2</b>
1.1. Historia de la teoría de grafos . . . . .	2
1.2. Noción de grafo . . . . .	8
1.3. Grado de un vértice . . . . .	10
1.4. Isomorfismo de grafos . . . . .	12
1.5. Subgrafos . . . . .	14
1.6. Conexidad en grafos . . . . .	15
1.7. Tipos de grafos simples . . . . .	19
1.8. Planaridad . . . . .	24
1.9. Coloración de grafos . . . . .	29
1.10. Matriz de adyacencia . . . . .	36
<b>2. Algoritmo de grafos</b>	<b>41</b>
2.1. Mathematica . . . . .	41
2.2. Algoritmo de la matriz de adyacencia . . . . .	44
2.3. Algoritmo de búsqueda de profundidad (DFS) . . . . .	46
2.4. Algoritmo de búsqueda de anchura (BFS) . . . . .	50
<b>3. Evasividad</b>	<b>55</b>
3.1. Árbol de decisiones y funciones evasivas . . . . .	55
3.2. La conjetura de evasividad (Aanderaa-Karp-Rosenberg) . . . . .	58
3.3. Aplicación en Mathematica . . . . .	61
<b>Conclusiones</b>	<b>65</b>
<b>Bibliografía</b>	<b>66</b>

# Introducción

El origen de la teoría de grafos se remonta a partir del trabajo de Leonhard Euler en el problema de los puentes Königsberg (1735), la cual dio el concepto de un grafo euleriano. La ciudad de Königsberg, antigua Alemania y ahora territorio ruso, estaba dividida en cuatro zonas por el río Pregel. Había siete puentes que comunicaban estas regiones; los habitantes de Königsberg deseaban encontrar un camino que cruzara por cada uno de los siete puentes sólo una vez y regresara al lugar de partida. Leonhard Euler publicó el artículo *Solutio problematis ad geometriam situs pertinentis* (*La solución de un problema referente a la geometría de posición*) en la Academia de Ciencias de San Petersburgo, Rusia, probando que no existe tal camino que pase por los siete puentes sólo una vez. Euler no dibujó un grafo con el fin de resolver el problema, pero reformuló el problema tratando de encontrar un camino entre las cuatro áreas de tierra, denotadas como  $A$ ,  $B$ ,  $C$  y  $D$ . A partir de este problema se origina algunos conceptos y percepciones de la teoría de grafos.

En el Capítulo 1 hablaremos un poco del origen de la teoría de grafos y su recorrido durante la historia, se tomarán algunos conceptos y definiciones; entre los más importantes en los que se encuentra el grafo simple, multigrafos, grafos dirigidos y no dirigidos, también se tomará en cuenta teoremas entorno a los grados de un vértice, como lo es el apretón de manos. Un concepto muy importante para este trabajo es el isomorfismo entre grafos, que permite observar si un par o más grafos tienen las mismas propiedades; además hablaremos de subgrafos y de la conexidad que también se abarcan propiedades y lemas. Se menciona una de sus grandes aplicaciones como lo es la coloración de grafos, de este se hablará el teorema de los cuatro colores que es más relevante y como concepto importante para este trabajo es la matriz de adyacencia; que permite representar un grafo por medio de una matriz.

Para el Capítulo 2 se realizó el algoritmo de la matriz de adyacencia, para este se utilizó el software matemático *Mathematica*; además de utilizar comandos entorno a la teoría de grafos, estos son el algoritmo de búsqueda de profundidad (DFS) y búsqueda de anchura (BFS).

En el último Capítulo, definiremos la conjetura de evasividad en grafos, para ello los conceptos de árbol de decisiones y funciones evasivas son fundamentales para el desarrollo de esta. Como manera de implementación desarrollamos un algoritmo realizado en *Mathematica*, que nos permite a modo de juego observar si una propiedad es o no evasiva.

Ya para finalizar daremos unas conclusiones obtenidas a lo largo de este trabajo de grado.

# Capítulo 1

## Elementos de la teoría de grafos

En este capítulo hablaremos de los orígenes, definiciones y resultados de la teoría de grafos dando algunos ejemplos y características de los mismos.

### 1.1. Historia de la teoría de grafos

Según [4] y como se vio en la introducción, el origen de la teoría de grafos se remonta a partir del trabajo de Leonhard Euler en el problema de los puentes Königsberg (1735), la cual dio el concepto de un grafo euleriano. La ciudad de Königsberg, antigua Alemania y ahora territorio ruso, estaba dividida en cuatro zonas por el río Pregel. Había siete puentes que comunicaban estas regiones; los habitantes de Königsberg deseaban encontrar un camino que cruzara por cada uno de los siete puentes sólo una vez y regresara al lugar de partida. Leonhard Euler publicó el artículo *Solutio problematis ad geometriam situs pertinentis* (*La solución de un problema referente a la geometría de posición*) en la Academia de Ciencias de San Petersburgo, Rusia, probando que no existe tal camino que pase por los siete puentes sólo una vez. Euler no dibujó un grafo con el fin de resolver el problema, pero reformuló el problema tratando de encontrar un camino entre las cuatro áreas de tierra, denotadas como  $A$ ,  $B$ ,  $C$  y  $D$  (véase en la Figura 1.1), de tal manera que los pares  $AB$  y  $AC$  son adyacente dos veces (correspondiente a los dos puentes entre  $A$  y  $B$  y entre  $A$  y  $C$ ), y los pares de  $AD$ ,  $BD$ , y  $CD$  son adyacentes sólo una vez (correspondiente a los puentes restantes). Euler demostró por un argumento de recuento que no existe tal camino, probando así que el problema de los puentes de Königsberg no tiene solución.

Las principales conclusiones de Euler para la situación general fueron las siguientes:

- Si hay más de dos áreas en la que se comunican por un número impar de puentes, tal camino es imposible de encontrar.
- Si el número de puentes es impar para exactamente dos áreas, el recorrido es posible si se inicia en cualquiera de estas dos áreas.
- Si, finalmente, no existen áreas a las que llevan un número impar de puentes, entonces el viaje requerido se puede lograr a partir de cualquier área.



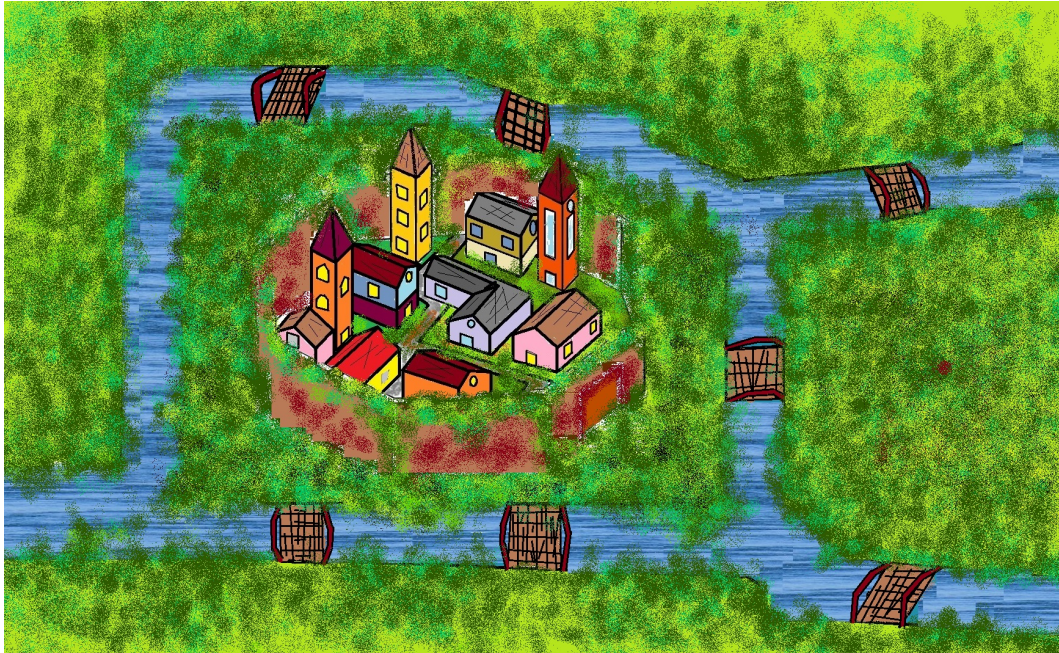


Figura 1.1: Puentes de Königsberg

Estos resultados corresponden a las condiciones bajo las cuales un grafo es euleriano. De igual manera, Euler también quiso observar lo contrario, que si se cumplen las condiciones anteriores, entonces una ruta es posible y dio una razón heurística del por qué esto debería ser así, pero no lo probó.

Un campo de estudio relacionado a esta teoría fue los rompecabezas de trazados, donde se puede elaborar un diagrama dado con el menor número posible de conexiones. En 1809 L. Poincaré escribió un libro sobre memorias en polígonos y poliedros, en el cual se plantea el siguiente problema:

*Considerando algunos puntos situados al azar en el espacio, se requiere disponer de un solo hilo flexible que une todos los puntos, dos en dos, de todas las maneras posibles, de modo que los dos extremos del hilo se unen y la longitud total es igual a la suma de todas las distancias mutuas.*

Poincaré observó que una solución sólo es posible cuando el número de puntos es impar, y dio un método para encontrar una disposición de este tipo para cada valor posible. En la terminología moderna, la cuestión tiene que ver con caminos de Euler en los grafos completos de orden impar. Luego, en el siglo XIX, se encontró la conexión entre el problema de los puentes de Königsberg y los rompecabezas de trazados dada por W. W. Rouse Ball, que parece haber sido el primero en utilizar el grafo de la Figura 1.2 para resolver el problema.

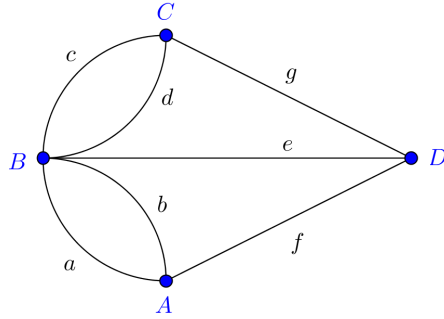


Figura 1.2: Grafo de Rouse Ball de los puentes de Königsberg

Un tipo de problema de grafos que se parece, superficialmente, al problema euleriano es el de encontrar un ciclo que pasa sólo una vez a través de cada vértice de un grafo dado. Por la influencia de Hamilton, estos grafos se llaman hamiltonianos, en lugar de ser más justamente el nombre de Kirkman, quien antes de la consideración de Hamilton del dodecaedro, consideró el problema más general.

En 1856 Kirkman escribió un artículo de investigación de los poliedros, para los que uno puede encontrar un ciclo que pasa por todos los vértices sólo una vez. Demostró que cada poliedro de caras con un número par de lados y un número impar de vértices no tiene tal ciclo, y ha puesto como ejemplo el poliedro obtenido por “cortar en dos la célula de una abeja” (véase en la Figura 1.3)

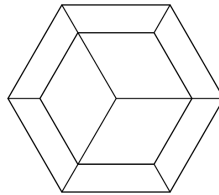


Figura 1.3: Ejemplo de “celda de la abeja” de Kirkman.

A partir de su trabajo en álgebra no conmutativa, Hamilton considera ciclos pasando por todos los vértices de un dodecaedro. Posteriormente, inventó un juego llamado el *Juego Icosian*, en el cual el jugador fue desafiado a encontrar tales ciclos que pasen por todos los vértices de un dodecaedro sólido y que el punto final sea el mismo punto inicial.

El concepto de árbol, un grafo conectado sin ciclos, apareció implícitamente en el trabajo de Gustav Kirchhoff (1824-87), quien empleó ideas de la teoría de grafos en el cálculo de corrientes en redes eléctricas. Más tarde, los árboles fueron utilizados por Arthur Cayley, James Joseph Sylvester, Georg Pólya y otros, en relación con la enumeración de ciertas moléculas químicas. Las técnicas de enumeración que involucran árboles surgieron por primera vez en relación con un problema en el cálculo diferencial, pero pronto llegaron a ser herramientas fundamentales en el conteo de moléculas químicas, así como proporcionar un fascinante tema de interés en su propio derecho.

Mientras que trabaja en un problema inspirado en algún trabajo de Sylvester en “Transformación diferencial y la reversión de series”, Cayley fue conducido a la enumeración de las raíces de los árboles. El método de Cayley consiste en tomar un árbol con raíz y luego retirarla, obteniendo de esta manera una serie de pequeños árboles con raíz cada uno, dejando  $A_n$  el número de árboles con raíces de  $n$  ramas Cayley demostró que la función generadora

$$1 + A_1x + A_2x^2 + A_3x^3 + \dots$$

es igual al producto

$$(1 - x)^{-1} \cdot (1 - x^2)^{-A_1} \cdot (1 - x^3)^{-A_2} \cdot \dots$$

Usando esta igualdad, él fue capaz de calcular los primeros números  $A_n$ , uno a la vez.

La fórmula de poliedros de Euler fue la base para la teoría de grafos topológicos, puesto que sostiene también para los grafos planares. Más tarde se amplió a superficies que no sean de la esfera. En 1930, una caracterización fundamental de grafos dentro de la esfera fue dada por Kazimierz Kuratowski, y el trabajo reciente – en particular por Neil Robertson, Paul Seymour, y otros – se ha extendido estos resultados a las superficies de orden superior.

Los griegos estaban familiarizados con los cinco sólidos regulares, pero no hay evidencia de que sabían la simple conexión entre el número de vértices  $V$ ,  $E$  de aristas, y  $F$  de las caras de un poliedro:

$$V - E + F = 2$$

En el siglo XVII, René Descartes estudió los poliedros y obtuvo resultados sobre la fórmula de Euler que más tarde se podría derivar. Sin embargo, puesto que Descartes no tenía el concepto de un borde, no podía hacer tal deducción. La primera aparición de la fórmula poliedro apareció en una carta, el 14 de noviembre de 1750, de Euler dirigida a Goldbach, donde indica el número de caras, ángulos sólidos (vértices) y las articulaciones (bordes) de  $\mathcal{H}$ ,  $\mathcal{S}$ , y  $\mathcal{A}$ , escribió:

*En cada sólido encerrado por caras planas al agregar el número de caras y el número de ángulos sólidos excede en dos el número de bordes, o  $\mathcal{H} + \mathcal{S} = \mathcal{A} + 2$ .*

Euler fue incapaz de demostrar su fórmula. La primera prueba válida fue dada por Legendre en 1794, utilizando las propiedades métricas de polígonos esféricos. En 1813, Cauchy obtuvo una prueba de la fórmula de Euler utilizando la proyección estereográfica, proyectando los poliedros en un plano y considerando una triangulación de los grafos planares resultantes; casi al mismo tiempo, Lhuillier dio una prueba topológica donde sólo hay cinco poliedros convexos regulares, y se anticipó a la idea de la dualidad señalando que cuatro de ellas se producen en pares recíprocos. También encontró tres tipos de poliedros en los que fallaron la fórmula de Euler, aquellos con muescas en sus caras, con una cavidad interior y poliedros en forma de anillo sobre un toro (es decir, poliedros con un ‘túnel’ a través de ellos). Para tales poliedros en forma de anillo, Lhuillier deduce la fórmula

$$V - E + F = 0$$

y extendió su discusión para demostrar que si  $g$  es el número de túneles en una superficie sobre la que se dibuja un mapa poliédrico, entonces

$$V - E + F = 2 - 2g.$$

El número  $g$  ahora se llama el *género de la superficie* y el valor de la cantidad de  $2 - 2g$  se llama la *característica de Euler*. En 1861-2, Listing escribió *Der Census räumliche Complex*, una extensa investigación sobre los complejos simpliciales y estudió cómo sus propiedades topológicas afectan a la generalización anterior de la fórmula de Euler. Este trabajo demostró ser influyente en el desarrollo posterior de la topología, en particular, H. Poincaré tomó las ideas de Listing en sus cartas de 1895-1904, que sentaron las bases de la topología algebraica.

El estudio de los grafos planares se originó en dos problemas recreativos que involucran el grafo completo  $\mathcal{K}_5$  y el grafo bipartito completo  $\mathcal{K}_{3,3}$ . Estos grafos (que se muestran en la Figura 1.4) son los obstáculos principales a la planaridad, como fue demostrado posteriormente por Kuratowski.

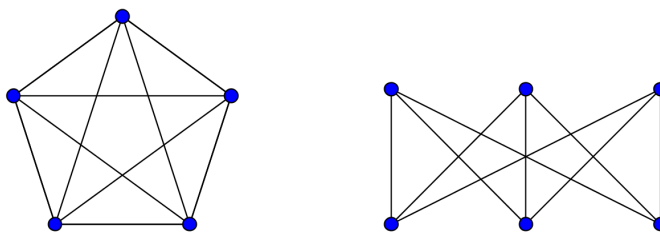


Figura 1.4: Los grafos de Kuratowski  $\mathcal{K}_5$  y  $\mathcal{K}_{3,3}$

Alrededor del año 1840, Möbius presentó el siguiente rompecabezas para sus estudiantes:

*Había una vez un rey con cinco hijos. En su testamento declaró que después de su muerte los hijos deben dividir el reino en cinco regiones tal que el límite de cada región debe tener una línea de frontera en común con cada una de las otras cuatro regiones. Los términos del testamento del rey pueden ser satisfechos.*

Esta cuestión se pregunta si se puede sacar cinco regiones vecinas entre sí en el plano. La conexión con la teoría de grafos se puede ver desde su doble versión, posteriormente formulada por H. Tietze:

*El rey afirmó además que los hijos deben unir las cinco capitales de sus reinos por caminos de modo que no hay dos caminos que se cruzan. ¿Se puede hacer esto?*

En esta doble formulación, el problema es el de decidir si el grafo  $\mathcal{K}_5$  es planar. Otro problema antiguo, es el problema de los servicios públicos, mencionado por H. Dudeney en el Strand Magazine de 1913:

*El problema es abastecer de agua, gas y electricidad, a partir de  $W$ ,  $G$  y  $E$ , a cada una de las tres casas,  $A$ ,  $B$ , y  $C$ , sin que ningún tipo de tubo se cruce de otra (como se muestra en la Figura 1.5)*

Este problema es el de decidir si  $\mathcal{K}_{3,3}$  es planar.

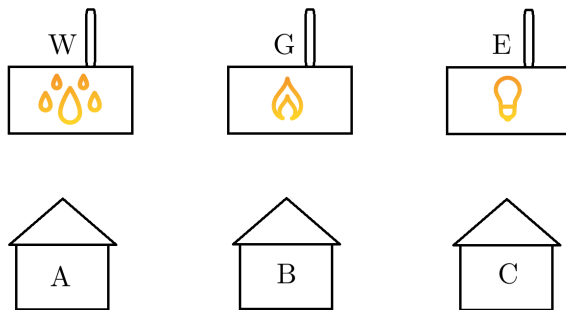


Figura 1.5: Problema de los servicios públicos

En 1930 Kuratowski publicó un artículo célebre que demuestra que todo grafo no planar tiene un subgrafo homeomorfo a  $\mathcal{K}_5$  o  $\mathcal{K}_{3,3}$ .

Los primeros trabajos sobre coloración se remontan a la coloración de los países sobre un mapa y, en particular, al famoso problema de los cuatro colores. *La mención más antigua conocida del problema de los cuatro colores se produce en una carta de A. De Morgan a Hamilton, el 23 de octubre de 1852.* De Morgan describió cómo un estudiante le había preguntado si cada mapa se puede colorear con sólo cuatro colores de tal manera que los países vecinos son de colores diferentes. El estudiante se identificó más tarde como Frederick Guthrie, dando crédito del problema a su hermano Francis, que lo formuló mientras coloreaba los condados de un mapa de Inglaterra. El 10 de abril de 1860, el problema apareció en la prensa, en una reseña de un libro sin firma en el Athenaeum (revista universitaria), escrita por De Morgan. Esta reseña se ha leído en los EE.UU. por C. S. Peirce, que desarrolló un interés de por vida en el problema. Una anterior referencia impresa, firmado por “F.G”, apareció en el Athenaeum en 1854. En 1879, Kempe, un antiguo estudiante de Cambridge de Cayley, publicó una supuesta prueba del teorema de cuatro colores en el American Journal of Mathematics. Kempe mostró que cada mapa debe contener un país con un máximo de cinco vecinos, y mostró cómo cualquier coloración del resto del mapa puede ampliarse para incluir un país. Su solución incluye una nueva técnica, conocida ahora como el argumento de cadena de Kempe, en el que se intercambian los colores de una sección de dos colores del mapa. La prueba de Kempe para ver un mapa que contiene un digon, triángulo o cuadrilátero era correcta, pero su argumento para el pentágono (donde utilizó dos colores-intercambios simultáneos) era errónea. En 1890, Heawood publicó un artículo en el Quarterly Journal of Pure and Applied Mathematics, señalando el error en la prueba de Kempe, salvando lo suficiente para deducir el teorema de cinco colores y generalizar el problema de varias maneras, tales como para otras superficies. Heawood publicó posteriormente otros seis artículos sobre el problema.



## 1.2. Noción de grafo

En esta sección comenzaremos como es usual con definiciones básicas para la construcción de grafos; en muchos textos están definidos de distintas maneras; en este caso se basará en las referencias [2], [3] y [4].

**Definición 1.** Un *Grafo*  $G = (V, E)$  consiste en dos conjuntos finitos  $V$  y  $E$  tal que:

- $V$  es no vacío y sus elementos son llamados *vértices* (*nodos*).
- Los elementos de  $E$  son llamados *aristas*. Cada arista tiene máximo dos vértices asociados a él. Es decir,  $u = \{a, b\} \in E$  para  $a, b \in V$ .

Sea  $G = (V, E)$  un grafo, si tomamos un par de vértices  $a, b \in V$  en el que  $u = \{a, b\} \in E$ , se dice que  $u$  *une* o *conecta* a  $a$  y  $b$ , luego  $a$  y  $b$  son *puntos adyacentes* a  $u$ , y se dice que estos vértices son *extremos* de dicha arista. También decimos que  $a$  y  $b$  son *incidentes* a la arista  $u$ , al igual que  $u$  es incidente a  $a$  y  $b$ . Si dos aristas  $u$  y  $v$  son adyacentes a un punto en común, entonces son *aristas adyacentes*.

Denotamos como  $\nu = |V|$  y  $\varepsilon = |E|$  como el número de vértices y aristas de un grafo respectivamente. Las notaciones de subíndices  $V_G$  y  $E_G$  se utilizan para denotar el conjunto de vértices y el conjunto de aristas, así como  $\nu(G)$  y  $\varepsilon(G)$  el número de vértices y de aristas respectivamente; cuando  $G$  no es el único grafo en consideración.

Los grafos se representan mediante diagramas en el plano de forma natural. Específicamente, cada vértice  $a \in V$  se representa por un punto o un círculo pequeño, y cada arista  $u \in E$  se representa por líneas.

**Ejemplo 1.** Sea  $G = (V, E)$  con  $V = \{a, b, c, d, e\}$  y  $E = \{\{a, b\}, \{a, c\}, \{b, c\}, \{b, e\}, \{c, d\}\}$ . En la Figura 1.6 se puede ver como se representa gráficamente el grafo, podemos ver que los vértices  $a$  y  $c$  son adyacentes y la arista  $u = \{a, c\}$  es incidente en dichos vértices.

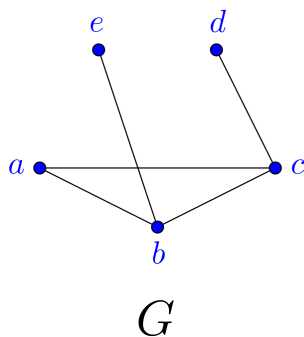


Figura 1.6: Representación gráfica de un grafo

Cuando dos aristas de un grafo se cruzan se dice que se *cortan* y llamamos *corte* al punto de intersección de ambas. En la Figura 1.6 las aristas  $u = \{a, c\}$  y  $v = \{b, e\}$  se cortan.

Si dos vértices están unidos por más de una arista, decimos que estas aristas son *aristas múltiples*. Una arista que tiene como extremo a un mismo vértice se denomina *lazo*. Con lo anterior, podemos decir que un grafo es simple cuando no tiene aristas múltiples ni lazos, esto lo formalizaremos en la siguiente definición.

**Definición 2.** Un grafo  $G = (V, E)$  es simple si para cualesquiera  $a, b \in V$  con  $a \neq b$ , existe solo una única arista  $u = \{a, b\} \in E$ .

También podemos decir que un grafo es simple, si el conjunto de aristas  $E$  es un conjunto de pares no ordenados distintos de  $V$ . Es decir, si para cualquier  $a, b \in V$  con  $a \neq b$ ,  $\{a, b\} = \{b, a\} \in E$ . Decimos que un grafo es *no simple* si tiene aristas múltiples o lazos. En la Figura 1.7 el grafo  $G$  es un grafo simple y el grafo  $H$  no lo es, ya que las aristas  $f$  y  $g$  son múltiples y la arista  $e$  es un lazo.

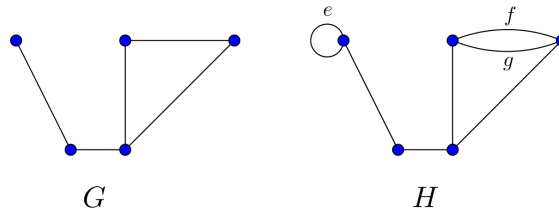


Figura 1.7: Ejemplo de grafos simples y no simples

Cuando se tiene un grafo simple  $G$  con  $n$  vértices, es interesante ver cuantas aristas puede tener dicho grafo. Esto es una característica que tiene los grafos simples y mostraremos a continuación.

**Proposición 1.** Sea  $G = (V, E)$  un grafo simple, entonces  $\varepsilon \leq \binom{\nu}{2}$ .

*Demostración.* Supongamos un grafo simple  $G = (V, E)$  con  $\nu$  vértices, cada vértice puede tener máximo  $\nu - 1$  vértices adyacentes, estos es  $\nu - 1$  aristas incidentes y si contamos todas las aristas por cada uno de los  $\nu$  vértices estaríamos contando cada arista dos veces. Luego  $2\varepsilon \leq \nu(\nu - 1)$ , así

$$\varepsilon \leq \frac{\nu(\nu - 1)}{2} = \binom{\nu}{2}.$$

□

Podemos decir que la Definición 2 es la definición estándar de un grafo. En los grafos simples se distinguen muchas clases, los cuales nos enfatizaremos en la Sección 1.7.

Los grafos simples no siempre van a ser útiles en ciertas situaciones en las cuales requiere la existencia de aristas múltiples, por lo que se recurren a los grafos no simples. A continuación, mostraremos la definición de multigrafo.

**Definición 3.** Un grafo  $G = (V, E)$  es un *multigrafo* si para cualesquiera  $a, b \in V$ ,  $a \neq b$ , existen aristas las cuales los unen. Es decir, un multigrafo permite aristas múltiples.

En comparación con los grafos simples, un grafo es un multigrafo si el conjunto de aristas  $E$  es un multiconjunto de pares no ordenados distintos de  $V$ . Observemos que la definición de multigrafo, no se admiten lazos. Porque podremos definir grafos más generales que los multigrafos.

**Definición 4.** Un *pseudografo* es un grafo que permite aristas múltiples y lazos.

Se puede decir que un multigrafo es un pseudografo sin lazos. En la Figura 1.8, el grafo de la izquierda es un multigrafo y el grafo de la derecha es un pseudografo.

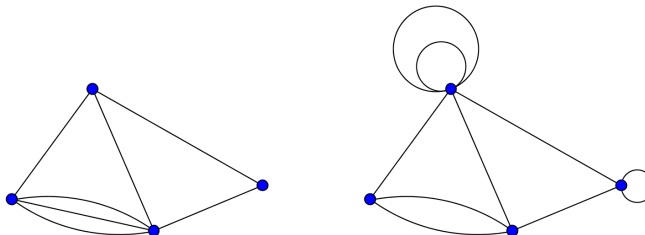


Figura 1.8: Un multigrafo y un pseudografo

Otro tipo de grafos son los dígrafos. Las aristas de estos grafos tienen dirección y son muy útiles en sistemas dinámicos como computadoras digitales y sistemas de flujo.

**Definición 5.** Un grafo es *dirigido* o *digrafo* si las aristas tienen una dirección, dichas aristas se denominan *aristas dirigidas*.

En los grafos simples, multigrafos y pseudografos, el conjunto de aristas  $E$  se conforma de pares de vértices no ordenados de  $V$ . En cambio, el conjunto de aristas  $E$  de los dígrafos es un conjunto de pares ordenados de elementos distintos de  $V$ .

Las aristas dirigidas son representadas en el plano como flechas. Un grafo en el que sus aristas no poseen un sentido definido, se dice que es un *grafo no dirigido*. En la Figura 1.9 el grafo de la izquierda es un grafo no dirigido, y el de la derecha es un dígrafo.

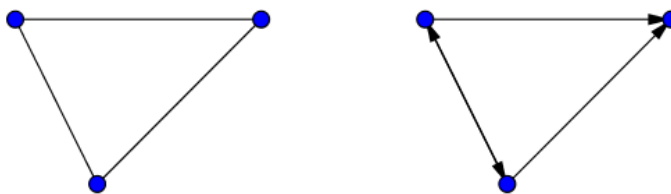


Figura 1.9: Ejemplo de dígrafo y grafo no dirigido

### 1.3. Grado de un vértice

En esta sección trabajaremos una de las nociones más fundamentales de la teoría de grafos, ya que este da cierta información del grafo al cual pertenece. A continuación, daremos la definición del grado de un vértice tomada de [3].



**Definición 6.** El *grado* un vértice  $a$  es igual al número de aristas que son incidentes al vértice, este es denotado como  $gr(a)$

Otra forma de obtener el grado de un vértice a parte de la definición anterior es por medio de su vecindad. La *vecindad* de un vértice, denotada como  $N(a)$  está conformado por los vértices adyacentes a  $a$ . Es decir,  $N(a) = \{b \in V | \{a, b\} \in E\}$ . De modo que  $gr(a) = |N(a)|$ .

Cabe resaltar que si el grado de un vértice es 0, decimos que es un *vértice aislado*, y si un vértice tiene un lazo, este se cuenta dos veces. Un par de características importantes en los grafos son definidas a continuación.

**Definición 7.** Sea  $G = (V, E)$  un grafo, el *grado máximo* es el máximo grado de todos los vértices del grafo, denotado por  $\Delta(G)$ . Análogamente, el *grado mínimo* es el mínimo grado de todos los vértices del grafo, denotado por  $\delta(G)$ .

Con los grados de todos los vértices de un grafo, obtendremos una forma de representarlo. A continuación, la definimos.

**Definición 8.** La *sucesión de grados* de un grafo  $G$  es la sucesión de los grados de los vértices de  $G$ , ordenada de manera creciente.

**Ejemplo 2.** En el grafo de la Figura 1.10, el grado máximo es de  $\Delta = 3$ , el grado mínimo es  $\delta = 1$  y la sucesión de grados es  $\{1, 2, 2, 3, 3, 3\}$ .

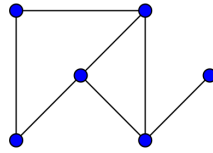


Figura 1.10: Grafo del Ejemplo 2

Resaltamos que no toda sucesión de  $n$  números es una sucesión de grados de un grafo con  $n$  vértices, por lo que hay ciertas condiciones para que ocurra, una de ellas es que el valor máximo de la sucesión sea menor al número de elemento de la sucesión.

Para la siguiente condición tomaremos el Ejemplo 2, si sumamos los elementos de la sucesión de grados da 14, exactamente el doble de la cantidad de aristas. Pero esto ocurre en cualquier grafo. Esta condición, es un resultado que relaciona la sumatoria de los grados de los vértices de un grafo con el número de aristas el cual se conoce como el *Teorema del apretón de manos*.

**Teorema 1.** La suma de los grados de los vértices de  $G$  es igual al doble del número de aristas de  $G$ . Es decir,

$$\sum_{a \in V} gr(a) = 2\varepsilon$$

*Demostración.* Sea  $G = (V, E)$  un grafo, supongamos que  $a \in V$  y  $gr(a) = s$ , esto indica que hay  $s$  aristas que inciden en  $a$ , entonces todas las  $s$  aristas inciden en un vértice  $b_i$  con  $i = 1, 2, \dots, s$ , así cada arista  $u = \{a, b_i\} \in E$ , de igual forma, es contada dos veces. Por lo tanto, hay  $2\varepsilon$  incidencias en el grafo, y esto se obtiene de la suma de los grados de los vértices.  $\square$

Si el grado máximo y el grado mínimo son el mismo, es decir,  $\Delta = \delta = k$ , la sucesión de grados sería  $\{k, k, \dots, k\}$ . Los grafos con estas características se definirán.

**Definición 9.** Un grafo  $G$  es *regular* si cada vértice tiene el mismo grado. Un grafo regular con vértices de grado  $k$  es llamado grafo  $k$ -regular.

Los grafos 0-regulares son grafos con vértices desconectados, Los grafos 1-regulares tienen las aristas desconectadas, y los grafos 2-regulares son ciclos o unión disjunta de ciclos, dicho término se habla en la Sección 1.6. Los grafos 3-regulares se conoce como grafo *cúbico*. Un ejemplo de estos grafos se puede observar en la Figura 1.11

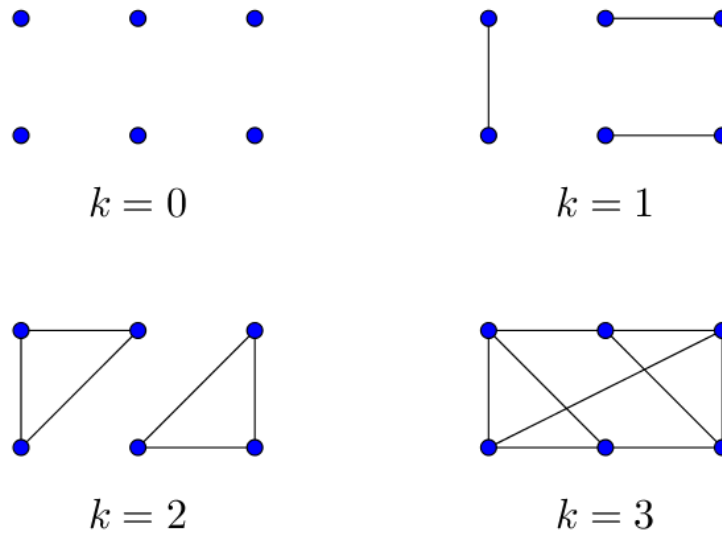


Figura 1.11: Ejemplos de grafos  $k$ -regulares con  $k = 0, 1, 2, 3$

## 1.4. Isomorfismo de grafos

En campos como la topología y las álgebras se maneja el término *isomorfismo*. Podemos encontrar grafos que a simple vista son muy diferentes pero conservan las mismas propiedades. Para aclarar lo dicho anteriormente, veamos un ejemplo similar al del texto [3]. Sea  $G$  un grafo con vértices  $V_G = \{1, 2, 3, 4, 5\}$  y aristas  $E_G = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 1\}\}$ , y sea  $H$  con vértices  $V_H = \{1, 2, 3, 4, 5\}$  y aristas  $E_H = \{\{1, 3\}, \{3, 5\}, \{5, 2\}, \{2, 4\}, \{4, 1\}\}$  como se muestra en la Figura 1.12.

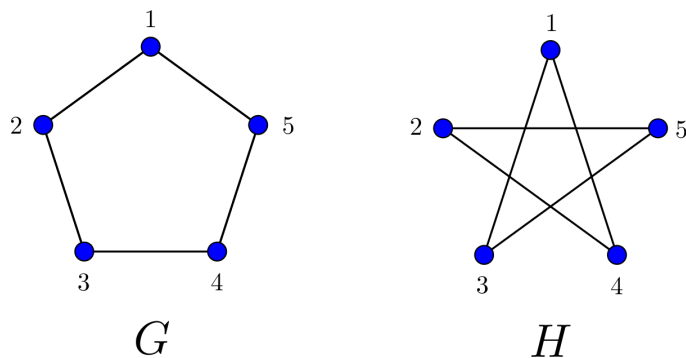


Figura 1.12: Grafos  $G$  y  $H$  isomorfos.

A pesar de no tener las mismas aristas, estos grafos son similares. Con la idea dejada en este ejemplo hablaremos a continuación de isomorfismo de grafos; en el cual se encuentran con más detalle en las referencias [3] y [4]. Antes de definir el isomorfismo de grafos, definimos  $r$  como una relación de adyacencia entre los vértices.

**Definición 10.** Sea  $G$  y  $H$  grafos. Un *isomorfismo* entre estos grafos es un par de biyecciones  $\phi = (\phi_V, \phi_E)$ , con  $\phi_V : V_G \rightarrow V_H$  y  $\phi_E : E_G \rightarrow E_H$ , tal que para cada par de vértices  $a, b \in V_G$ , el subconjunto de aristas de  $E_G$  que unen a  $a$  y  $b$  se correlaciona biyectivamente con el subconjunto de aristas de  $E_H$  que unen a los vértices  $\phi(a)$  y  $\phi(b)$ .

Decimos que  $G$  y  $H$  son *isomorfos* y lo denotamos  $G \cong H$  si hay un isomorfismo  $\phi : G \rightarrow H$ .

En otras palabras, un isomorfismo es una biyección entre el conjunto de vértices la cual preserva la adyacencia. La anterior definición, es la generalización de los isomorfismos de grafos, por lo que se debe poner condiciones en la definición para cada tipo de grafo, la siguiente es la definición de isomorfismo para grafos simples.

**Definición 11.** Sea  $G$  y  $H$  grafos simples. Un *isomorfismo* entre estos grafos es una biyección entre el conjunto de vértices  $\phi : V_G \rightarrow V_H$  tal que  $\{a, b\} \in E_G$  si y solamente si  $\{\phi(a), \phi(b)\} \in E_H$ , con  $a, b \in V_G$ , en términos de la relación un isomorfismo entre  $G$  y  $H$  es una biyección  $\phi : V_G \rightarrow V_H$  tal que para  $a, b \in V_G$ ,  $a r b$  si y solo si  $\phi(a) r \phi(b)$

Para los grafos de la Figura 1.12. podemos encontrar una biyección  $\phi$  tal que  $\phi(1) = 1$ ,  $\phi(2) = 3$ ,  $\phi(3) = 5$ ,  $\phi(4) = 2$  y  $\phi(5) = 4$ . En general, no es sencillo verificar si dos grafos son isomorfos o no. Si dos grafos son isomorfos, fácilmente se puede encontrar la biyección por medio de su dibujo, sin importar que los dibujos de los dos grafos sean muy diferentes. En conclusión, podemos ver que si dos grafos son isomorfos, conservan las propiedades, con esto damos paso a la siguiente definición.

**Definición 12.** Un *invariante* de un grafo  $G$  es un número asociado al grafo que tiene el mismo valor para cualquier grafo isomorfo a  $G$ .

El número de vértices, el número de aristas, y los grados de los vértices, son ejemplos de invariantes. Pero dos grafos con el mismo número de vértices y el mismo número de

aristas no tienen por qué ser isomorfos. Un ejemplo son los grafos de la Figura 1.13; cada grafo tienen seis vértices, cinco aristas y su sucesión de grados es  $(1, 1, 1, 2, 2, 3)$ , pero estos no son isomorfos, ya que, en el primer grafo, el vértice de grado 3 es adyacente a dos de grado 1 y de uno de grado 2; y en el vértice de grado 3 del otro grafo, es adyacente a uno de grado 1 y de dos de grado 2.

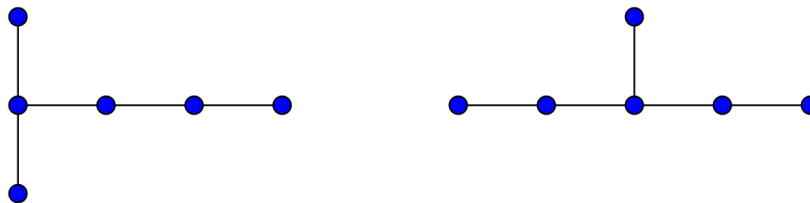


Figura 1.13: Ejemplo de grafos no isomorfos

## 1.5. Subgrafos

En muchos casos podemos sacar información de un grafo por medio de otros grafos que están incluidos en el, en esta sección trabajaremos el término subgrafo tomada de la referencia [1].

**Definición 13.** Sea  $G = (V_G, E_G)$  y  $H = (V_H, E_H)$  grafos, decimos que  $H$  es un *subgrafo* de  $G$  si el conjunto de vértices es un subconjunto del conjunto de vértices de  $G$ , es decir,  $V_H \subseteq V_G$ , y  $E_H$  contiene todas las aristas de  $G$  que unen los vértices de  $V_H$ .

Con la definición anterior, surge un par de subgrafos que son relevantes. Si tenemos dos grafos  $G = (V_G, E_G)$  y  $H = (V_H, E_H)$ , diremos que  $H$  es un subgrafo *completo* de  $G$  si  $E_H$  contiene todas las aristas de  $E_G$  que unen los vértices de  $V_H$ ; y diremos que  $H$  es un subgrafo *extendido* de  $G$  si  $\nu(H) = \nu(G)$ , es decir, tiene todos los vértices de  $G$ . Como ejemplo tenemos la Figura 1.14, vemos tres subgrafos del grafo  $G$ , el subgrafo  $G_1$ , el subgrafo completo  $G_2$  y el subgrafo extendido  $G_4$ .

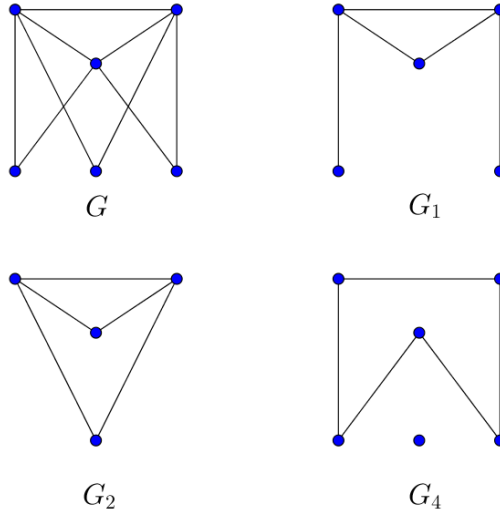


Figura 1.14: Subgrafos del grafos  $G$

Los subgrafos que conservan casi todas las propiedades de un grafo son los mas interesantes en estudiar. Por lo que es posible modificar mínimamente un grafo. Sea  $G$  un grafo, *eliminar una arista  $u$*  es quitarla del conjunto de aristas  $E$ , obteniendo un subgrafo de  $G$  denotado por  $G - u$ , que es expansivo. *Adicionar una arista  $u$*  es el resultado de añadir una arista al conjunto  $E$  tal que una dos vértices cualesquiera ya existentes en el grafo, y se denota por  $G + u$ . Observemos que  $(G - u) + u = G$ . Un ejemplo de estos subgrafos están en la Figura 1.15.

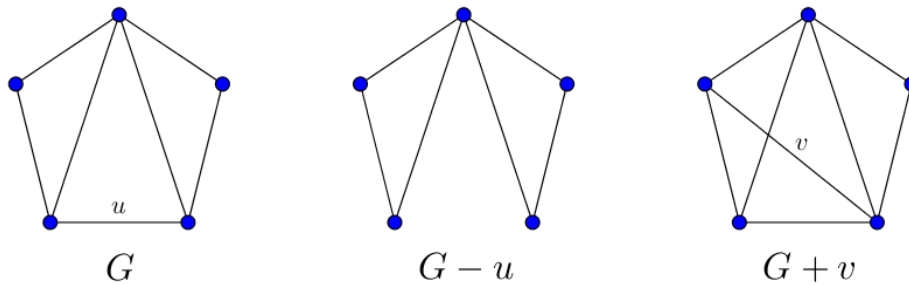


Figura 1.15: Ejemplo de adicción y eliminación de aristas

## 1.6. Conexidad en grafos

Un concepto básico en la teoría de grafos es la conexidad. Para desarrollar esta idea definiremos primero las distintas maneras de moverse por un grafo, de vértice a vértice y siguiendo las aristas; así como se puede apreciar en [3] y [4].

**Definición 14.** Un *paseo* de un grafo  $G$  es una sucesión finita de vértices  $x_0, x_1, \dots, x_r$  (pueden repetirse vértices) de forma que  $\{x_i, x_{i+1}\}$  es una arista de  $G$ , para cada  $i = 0, 1, \dots, r-1$ . Podemos denotar como  $W = x_0u_1x_1u_2x_2 \cdots x_{k-1}u_kx_k$  al paseo de longitud  $k$

Podemos decir que el paseo conecta  $x_0$  y  $x_r$ , y que la longitud del paseo será  $r$ , que puede ser menor o igual al número de vértices visitados.

Un paseo  $x_0, \dots, x_r$  se dice que es *cerrado* si  $x_0 = x_r$ . También decimos que un paseo es un *camino* si no se repite ninguna arista, si un camino no se repite ningún vértice se dice que es un *camino simple*, y si un camino es cerrado se le llama *circuito*. Decimos que un *ciclo* es un paseo cerrado  $x_0, \dots, x_r$ , con  $r \geq 3$  cuyos vértices sean todos distintos excepto el primero y el último.

En el grafo de la Figura 1.16, podemos encontrar un camino que conecta a los vértices  $u$  y  $v$  pasando por los vértices restantes. Pero, este camino contiene un ciclo. Observemos que al quitarlo no supone ningún problema, ya que nos quedamos con un camino que sigue conectando los vértices extremos; aunque, por supuesto, la longitud del camino cambiará.

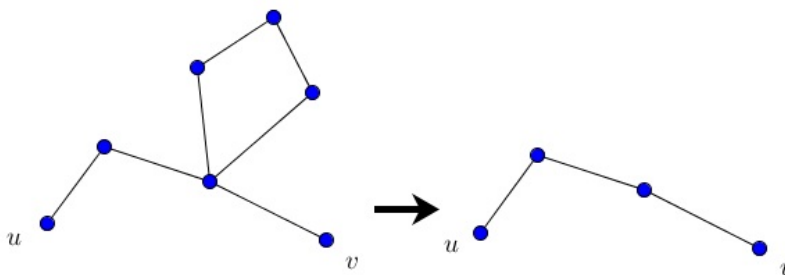


Figura 1.16: Camino que conecta los vértices  $u$  y  $v$

Basándonos en lo dicho anteriormente, definiremos la conexidad en grafos.

**Definición 15.** Un grafo  $G$  es **conexo** si dados cualesquiera dos vértices distintos  $a, b \in V$ , podemos encontrar un paseo que los conecte.

Podemos ver que la definición abarca a los grafos con al menos dos vértices. Por tanto, asumiremos que un grafo con un único vértice es también conexo. Observemos que si existe un paseo que conecte dos vértices  $a$  y  $b$ , entonces podemos encontrar un camino que los conecte. Más aún, podremos encontrar un camino simple, cuya longitud no podrá ser mayor que  $\nu(G) - 1$ . Así que, en lo que sigue, a veces manejaremos la noción de conexión por caminos.

Lo más interesante ahora es determinar qué ocurre cuando un grafo no es conexo. Por intuición en un grafo no conexo, hay vértices que no pueden ser conectados, por lo que el grafo estará formado por diversos bloques de vértices, cada uno de los cuales será un grafo conexo.

Sea  $G = (V, E)$  un grafo y  $a \in V$ , llamaremos *componente conexa de  $a$  en  $G$*  denotado por  $C_a$ , al subgrafo con conjunto de vértices con los que existe un camino con el vértice  $a$ . Resaltamos que si  $a$  está en la componente de  $b$ , entonces  $b$  está en la de  $a$ . Ahora, ¿qué sucede con las componentes conexas de dos vértices?. Con esta pregunta damos paso al siguiente teorema.

**Teorema 2.** *Sea  $a, b$  vértices de un grafo  $G = (V, E)$ , las componentes conexas de dichos vértices, o bien son las mismas o bien son disjuntas.*

*Demostración.* Sea  $a, b \in V$ , con sus componentes conexas  $C_a$  y  $C_b$  respectivamente. Podemos ver dos casos que puede ocurrir, que exista un camino entre  $a$  y  $b$  y que no exista tal camino.

Si existe un camino entre  $a$  y  $b$ . Tomamos un vértice  $c \in C_a$  entonces existe un camino entre  $c$  y  $a$ . Si  $b$  esta en dicho camino, entonces  $b$  esta en  $C_a$ . Si  $b$  no está en el camino, construimos un camino uniendo el camino de  $c$  y  $a$ , y el de  $a$  y  $b$ , por lo que  $b$  esta en  $C_a$ . Así,  $C_a = C_b$ .

Si no existe un camino entre  $a$  y  $b$ . Supongamos que la intersección entre  $C_a$  y  $C_b$  no es vacía, entonces existiría un vértice  $c$  que está en  $C_a$  y  $C_b$ , entonces existe un camino entre  $a$  y  $c$ , y existe un camino entre  $b$  y  $c$ , y con estos encontramos que existe un camino entre  $a$  y  $b$ , lo cual es una contradicción de lo supuesto. Así,  $C_a \cap C_b = \emptyset$ .  $\square$

Con el anterior teorema, podemos dar paso a la siguiente definición.

**Definición 16.** Sea  $G = (V, E)$  un grafo, una *componente conexa* de  $G$  es un subgrafo conexo más grande contenido en  $G$ .

Otra forma de ver una componente conexa es tomando todos los vértices que está en la componente conexa de un cierto vértice de  $V$  y todas las aristas del grafo que unen estos vértices. Observemos que las componentes conexas de un grafo son grafos conexos y claramente vemos que un grafo se puede representar como unión de grafos conexos, que a la vez son sus componentes conexas.

En la sección anterior hablamos de eliminar una arista  $u$  de un grafo  $G$  obteniendo el subgrafo  $G - u$ , La siguiente definición relaciona esto con las componentes conexas de grafo.

**Definición 17.** Sea  $u$  una arista de un grafo  $G = (V, E)$ , diremos que es un *puente* si el subgrafo  $G - u$  tiene una componente conexa de más que el grafo  $G$ .

Una forma equivalente de identificar un puente la podemos ver en el siguiente resultado.

**Proposición 2.** Sea  $G$  un grafo, una arista  $u \in E$  es un puente si y sólo si  $u$  no pertenece a un ciclo de  $G$ .

*Demostración.* Sea  $u \in E$ , sabemos que el subgrafo  $G - u$  tiene una componente conexa de más que  $G$ . Si  $u$  pertenece a un ciclo  $C$  y tiene extremos  $x_0, x_1 \in V$ , sin pérdida de generalidad, tomamos a  $C = x_0 u x_1 u_2 x_2 \cdots x_{k-1} u_k x_0$  para algún  $k$ . Así,  $C - u$  sería un camino simple  $x_1 u_2 x_2 \cdots x_{k-1} u_k x_0$ . Teniendo en cuenta que  $C - u \subset G - u$ , y por tanto,  $x_0$  y  $x_1$  están unidos por un camino en  $G - u$ . Como  $x_0$  y  $x_1$  son extremos de  $u$ , al añadir  $u$  a  $G - u$  no disminuye el número de componentes conexas porque  $x_0$  y  $x_1$  ya están unidos. Esto genera una contradicción. Por lo cual  $u$  no pertenece a ningún ciclo de  $G$ .

Recíprocamente, si una arista  $u \in E$  con extremos  $x_0$  y  $x_1$  no pertenece a ningún ciclo de  $G$  se tiene que los extremos de  $u$  están unidos por un único camino, que es precisamente  $x_0 u x_1$ . Así,  $G - u$  no tiene tal camino y, por tanto, tiene una componente conexa más que  $G$ , por tanto  $u$  es un puente de  $G$ .  $\square$

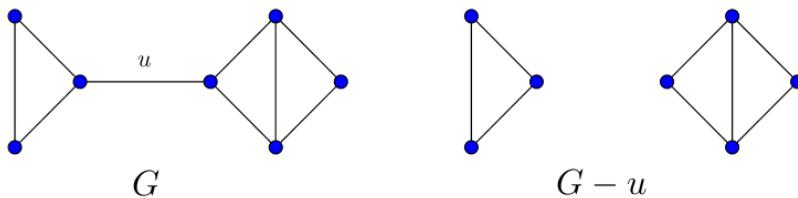


Figura 1.17: Ejemplo de un puente

Un grafo puede tener varios puentes. En la Figura 1.17 la arista  $u$  es el único puente del grafo  $G$  ya que el subgrafo  $G - u$  tiene más componentes conexas, exactamente dos, esto se da porque el grafo es conexo. Con este caso particular damos el siguiente resultado.

**Lema 1.** *Sea  $G$  un grafo conexo, si  $u$  es un puente de  $G$ , entonces  $G - u$  tiene exactamente dos componentes conexas.*

*Demostración.* Sea  $u \in E$  el puente de  $G$ . Tomemos los extremos de  $u$ ,  $a, b \in V$ , y dividimos el conjunto de vértices en dos. El conjunto  $V_1$  donde están los vértices que se conectan con  $a$  por un camino sin usar la arista  $u$ , y el conjunto  $V_2$  donde están los vértices que necesariamente usan la arista  $u$  para conectarse por un camino con  $a$ . Claramente,  $a \in V_1$  y  $b \in V_2$ . Ahora, suponemos  $V_1$  y  $V_2$  no son disjuntos, tomamos un vértice  $c \in V_1 \cap V_2$  entonces al grafo  $G - u$  sería conexo ya que  $c \in V_1$  y existe un camino entre  $c$  y  $a$ . Así,  $u$  no es un puente, lo cual es una contradicción de lo supuesto. Luego  $V_1$  y  $V_2$  son disjuntos, y son los conjuntos de vértices de las componentes conexas de  $G - u$   $\square$

Con el lema anterior, podemos establecer un resultado sobre grafos conexos.

**Proposición 3.** Si  $G$  es un grafo conexo, entonces  $\varepsilon \geq \nu - 1$

*Demostración.* Utilizaremos el principio de inducción fuerte sobre  $\varepsilon$ .

Si  $G$  no tiene aristas, entonces  $\varepsilon = 0$ , y para que el grafo sea conexo, debe tener un vértice. Es decir,  $\nu = 1$ . Si  $\varepsilon = 1$ , entonces, para que  $G$  sea conexo, debe tener dos vértices. Es decir,  $\nu = 2$ . Por ambos casos,  $\varepsilon = \nu - 1$ .

Supongamos cierto que si tenemos un grafo conexo con  $\varepsilon(G) = k$  aristas, para cualquier  $k \leq m$ , entonces  $\varepsilon(G) \geq \nu(G) - 1$ . Consideremos entonces un grafo conexo con  $\varepsilon(G) = m + 1$  aristas. Sea  $u$  una arista cualquiera de  $G$ , quitando la arista obtendremos el subgrafo  $H = G - u$ , este grafo tiene los mismos vértices de  $G$  pero con una arista menos. Es decir,  $\nu(H) = \nu(G)$  y  $\varepsilon(H) = \varepsilon(G) - 1$ . Caben dos posibilidades para el subgrafo  $H$ :

1. Si  $H$  sigue siendo conexo, por hipótesis de inducción,  $\varepsilon(H) \geq \nu(H) - 1$  y así

$$\begin{aligned} \varepsilon(H) &\geq \nu(H) - 1 \\ \varepsilon(G) - 1 &\geq \nu(G) - 1 \\ \varepsilon(G) &\geq \nu(G) \geq \nu(G) - 1 \end{aligned}$$



2. Si  $H$  no es conexo, entonces tendrá dos componentes conexas  $H_1$  y  $H_2$ , ambas son conexas y tienen menos aristas que  $G$ . Además,  $\varepsilon(H_1) + \varepsilon(H_2) = \varepsilon(H) = \varepsilon(G) - 1$  y  $\nu(H_1) + \nu(H_2) = \nu(H)$ . Por hipótesis de inducción.  $\varepsilon(H_1) \geq \nu(H_1) - 1$  y  $\varepsilon(H_2) \geq \nu(H_2) - 1$ . Luego,

$$\begin{aligned}\varepsilon(H_1) + \varepsilon(H_2) &\geq \nu(H_1) - 1 + \nu(H_2) - 1 \\ \varepsilon(H) &\geq \nu(H) - 2 \\ \varepsilon(G) - 1 &\geq \nu(G) - 2 \\ \varepsilon(G) &\geq \nu(G) - 1\end{aligned}$$

□

## 1.7. Tipos de grafos simples

En los grafos simples encontramos tanta variedad de estos grafos que sería muy útil clasificarlos dependiendo del número de vértices, número de aristas y demás propiedades que puedan cumplir. En esta sección hablaremos de cada tipo de grafo que se recopilieron de [1], [3] y [4].

**Definición 18.** Un grafo simple con  $n$  vértices se dice *vacío* si no contiene ninguna arista.

Si un grafo vacío con un solo vértice se dice que es un grafo *nulo*. En la sección 1.6 hablamos de camino simple y de ciclos, adaptaremos estas definiciones como tipos de grafos simples.

**Definición 19.** Un grafo simple con vértices  $V = \{a_1, a_2, \dots, a_n\}$  es *lineal* ( $n \geq 2$ ) si los vértices  $a_{i-1}$  y  $a_i$  están conectados por una sola arista, para  $i = 1, \dots, n$ . El grafo lineal con  $n$  vértices lo notamos como  $L_n$ .

Un grafo lineal con  $n$  vértices tiene  $n - 2$  vértices de grado 2, y los otros 2 tienen grado uno que son los extremos del grafo.

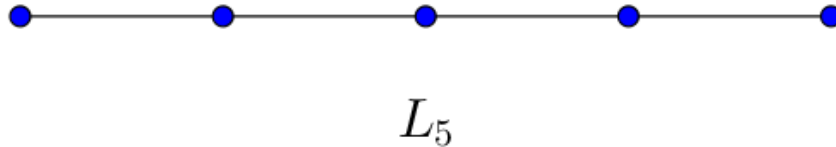


Figura 1.18: Ejemplo de grafo lineal

**Definición 20.** Sea  $G = (V, E)$  un grafo con el conjunto de vértices  $V = \{v_1, v_2, \dots, v_n\}$  con  $n \geq 3$ , dicho grafo es un *ciclo* si el conjunto de aristas sería  $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$ . El ciclo con  $n$  vértices lo denotaremos como  $\mathcal{C}_n$ .

Observemos que en cualquier grafo ciclo todos los vértices tienen grado 2, por lo que todos los grafos ciclos son 2-regular. Si al grafo ciclo  $\mathcal{C}_n$  le agregamos un vértice que conecta a los otros vértices del grafo, diremos que es una *rueda* y lo denotaremos como  $\mathcal{W}_n$ .

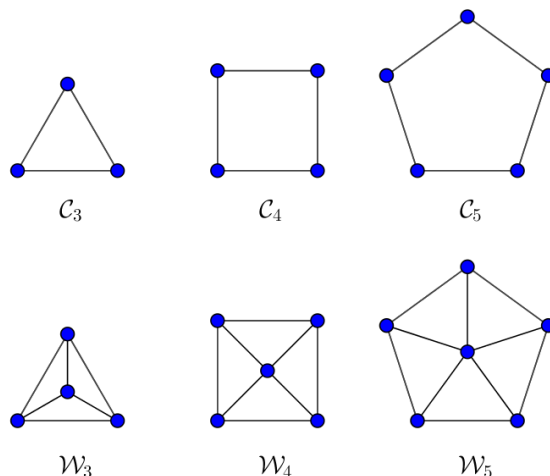


Figura 1.19: Los ciclos y ruedas para  $n = 3, 4, 5$

Por la Proposición 1, cualquier grafo simple con  $n$  vértices tiene máximo  $\binom{n}{2}$  arista. A continuación, hablaremos de los grafos simples que tienen exactamente  $\binom{n}{2}$  aristas.

**Definición 21.** Sea  $G = (E, V)$  un grafo simple, si para todo  $a, b \in V$  existe una arista  $u \in E$  tal que  $u = \{a, b\}$  dicho grafo se denomina *grafo completo*. Denotaremos como  $\mathcal{K}_n$  al grafo completo de  $n$  vértices.

Es evidente que si tenemos un grafo completo con  $n$  vértices tendría exactamente  $\frac{n(n-1)}{2}$  aristas y cada vértice tiene  $n - 1$  aristas incidentes, implicando que el grafo completo  $\mathcal{K}_n$  es un grafo  $(n - 1)$ -regular.

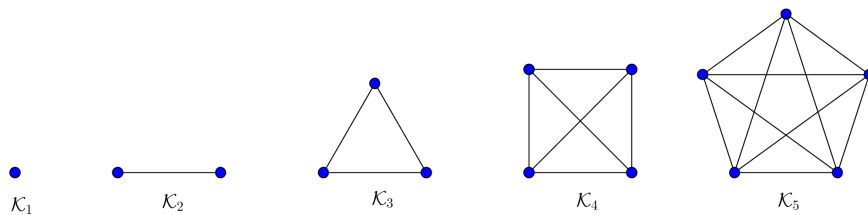


Figura 1.20: Los grafos completos para  $n = 1, 2, 3, 4, 5$

Los grafos mencionados anteriormente tiene la peculiaridad de que si dos grafos con cualquiera de eso tipos y con igual número de vértices, serán isomorfos. A continuación, definiremos otro tipos de grafos que, cuando dos grafos tenga el mismo número de vértices y el mismo tipo, no serán isomorfos.

**Definición 22.** Un grafo  $G = (V, E)$  es llamado *grafo bipartito* si el conjunto de vértices tiene una partición en dos conjuntos  $U$  y  $W$  de manera que las aristas sólo pueden conectar vértices de  $U$  con vértices de  $W$ .

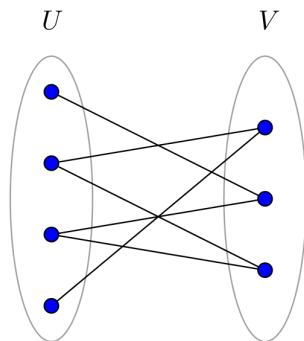


Figura 1.21: Ejemplo de grafo bipartito

Los grafos bipartitos suelen ser graficados en dos columnas o filas de vértices y las aristas uniendo los vértices de cada conjunto, como se puede apreciar en la Figura 1.21. Si los dos subconjuntos de vértices tiene la misma cantidad de vértices, decimos que el grafo  $G = (V, E)$  es *balanceado*.

Podemos deducir que un ciclo con un número impar de vértices no es bipartito, y si el ciclo tiene un número par de vértices es bipartito.

Un grafo bipartito con conjunto de vértices  $V = U \cup W$  en la cual cada vértice de  $U$  es adyacente a todos los vértices de  $W$ , es llamado *grafo bipartito completo*. Si un grafo bipartito completo tiene el conjunto de vértices  $V = U \cup W$ , con  $n$  el número de vértices de  $U$  y  $m$  el número de vértices de  $W$ , lo denotaremos como  $\mathcal{K}_{n,m}$ .

Es fácil ver que un grafo bipartito  $\mathcal{K}_{n,m}$  es igual al grafo bipartito  $\mathcal{K}_{m,n}$ . Además, el número de vértices de un grafo bipartito completo es  $m + n$  y el número de aristas es  $mn$ .

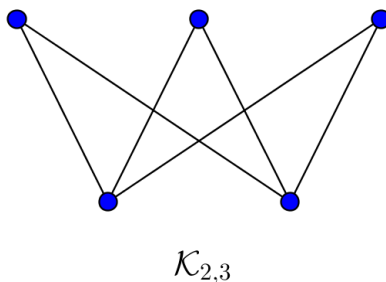


Figura 1.22: El grafo bipartito completo  $\mathcal{K}_{2,3}$

A continuación, mostraremos un tipo de grafo que, a pesar de su simplicidad tiene muchas aplicaciones en campos diferentes, como el diseño de algoritmos, problemas

de optimización, hasta en la misma teoría de grafos. Estos grafos son conocidos como árboles.

**Definición 23.** Sea  $G = (V, E)$  un *árbol* es un grafo conexo y sin ciclos.

Un ejemplo sencillo es un árbol son los grafos lineales  $L_n$ , también decimos que un grafo es un *bosque* si sus componentes conexas son árboles. En la Figura 1.23 mostramos otros ejemplos de árboles. En los árboles que vemos en la Figura 1.23 hay un vértice especial que origina todas las conexiones con los demás vértice, Por lo general, este vértices siempre se ubica a un extremo del árbol. Este vértice lo llamamos *raíz*.

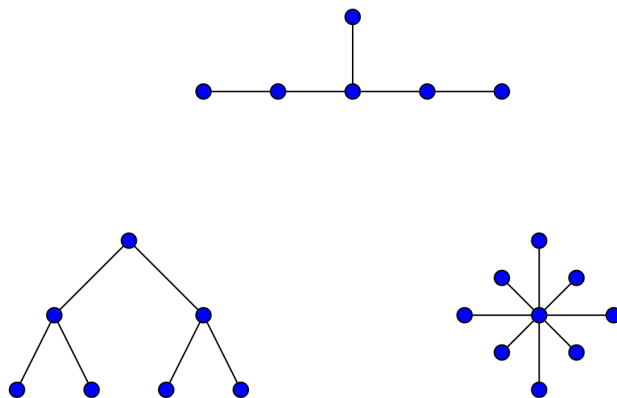


Figura 1.23: Ejemplos de árboles

La siguiente proposición nos muestra otra forma alternativa de definir un árbol.

**Proposición 4.** Un grafo es un árbol si y sólo si todo par de vértices de  $G$  está unido por un solo camino.

*Demostración.* Sea  $G$  un árbol. Por definición  $G$  es conexo, entonces existe un camino simple entre todo par de vértices de  $G$ . Si  $a$  y  $b$  son dos vértices cualesquiera de  $G$  y suponemos que existen dos caminos distintos de  $a$  a  $b$ , al unir dichos caminos se formaría un subgrafo de  $G$  que contendría un ciclo debido a que existen dos caminos diferentes, hecho que nos lleva a una contradicción a la definición de árbol. Por lo tanto, existe un solo camino entre  $a$  y  $b$ . Recíprocamente, si todo par de vértices de  $G$  está unido por un solo camino simple, obviamente  $G$  es un grafo conexo; si existiera un ciclo en  $G$  entonces podríamos encontrar dos caminos simples que unen a cualquier par de vértices, lo cual contradice la hipótesis. Por lo tanto no hay ciclos en el grafo  $G$ , que es lo mismo decir que el grafo  $G$  es un árbol.  $\square$

La definición de árbol no recoge una de las características fundamentales de los árboles, que son los conexos con el menor número de aristas que podemos tener. Las proposiciones 5 y 6 nos proporcionan caracterizaciones alternativas que recogen esta idea.

**Proposición 5.** Un grafo  $G$  es un árbol si y sólo si no tiene ciclos y, si añadimos una arista cualquiera, se forma un ciclo.

*Demostración.* Sea  $G$  un árbol, entonces es un grafo conexo y sin ciclos. Consideremos dos vértices cualesquiera que no estén unidos por arista alguna en  $G$ . Por estar en un grafo conexo, existirá un camino que los conecte en  $G$ . Al añadir una arista entre los vértices, se formará un ciclo.

Recíprocamente, sea  $G$  un grafo sin ciclos para el que añadir una arista cualquiera supone la formación de un ciclo. Supongamos que no fuera conexo. En este caso, al menos existirían dos vértices que no podríamos conectar en  $G$ . Pero entonces todavía podríamos añadir la arista que los une sin que se nos formara un ciclo, lo cual es una contradicción de la hipótesis. Luego  $G$  debe ser conexo, implicando que  $G$  sea un árbol.  $\square$

**Proposición 6.** Un grafo  $G$  es un árbol si y sólo si es conexo y tiene la propiedad de que al eliminar una arista cualquiera del grafo, este deja de ser conexo.

*Demostración.* Sea  $G$  un árbol, entonces es un grafo conexo y sin ciclos, y sea  $u$  un arista de  $G$  con extremos  $a$  y  $b$ , tal que, eliminándola, obtendríamos el grafo  $G - u$ . Si  $G - u$  fuera conexo, podríamos conectar en  $G - u$  los vértices  $a$  y  $b$ . Pero añadiendo la arista  $u$ , lo que tendríamos sería un ciclo en  $G$ , lo cual es una contradicción ya que  $G$  es un árbol. Así que  $G - u$  es no conexo.

Supongamos que  $G$  es un grafo conexo tal que para toda arista  $u$  de  $G$ ,  $G - u$  no es conexo. Si el grafo contuviera un ciclo, siempre podríamos quitar una arista de ese ciclo sin que el grafo dejara de ser conexo, lo que es una contradicción con la hipótesis. Luego, el grafo  $G$  no tiene ciclos, esto implica que  $G$  es un árbol.  $\square$

En la proposición 3 vemos que para cualquier grafo conexo  $G$ ,  $\varepsilon \geq \nu - 1$ . En los árboles llegamos a la igualdad, como se demostrará a continuación.

**Proposición 7.** Un grafo  $G$  es un árbol si y sólo si es conexo y  $\varepsilon = \nu - 1$

*Demostración.* Suponemos que  $G$  es un árbol. Utilizaremos el principio de inducción fuerte sobre el número de aristas  $\varepsilon$ .

Si  $G$  es un árbol con una arista, es decir,  $\varepsilon = 1$ , sólo cabe la posibilidad de que sea un  $L_2$ , para el que  $\nu = 2$  y así  $\varepsilon = \nu - 1$ .

Supongamos cierto que para todo árbol con  $\varepsilon(G) \leq d$  se tiene que  $\varepsilon(G) = \nu(G) - 1$ . Ahora consideremos un árbol cualquiera  $G$  con  $\varepsilon(G) = d + 1$ . Si fijamos una arista  $u$  de  $G$ , el subgrafo  $G - u = G_1 \cup G_2$  donde  $G_1$  y  $G_2$  son árboles con número de aristas  $\varepsilon(G_1) \leq d$  y  $\varepsilon(G_2) \leq d - 1$ . Además,  $\varepsilon(G_1) + \varepsilon(G_2) = d$ . Por hipótesis de inducción  $\varepsilon(G_1) = \nu(G_1) - 1$  y  $\varepsilon(G_2) = \nu(G_2) - 1$ . Así

$$\begin{aligned}\varepsilon(G_1) + \varepsilon(G_2) &= \nu(G_1) - 1 + \nu(G_2) - 1 \\ \varepsilon(G) - 1 &= \nu(G) - 2 \\ \varepsilon(G) &= \nu(G) - 1\end{aligned}$$

Recíprocamente, sea  $G$  un grafo conexo tal que  $\varepsilon(G) = \nu(G) - 1$ . Si contuviera un ciclo, podríamos quitar una arista  $u$  de ese ciclo sin que el grafo se desconectara. Pero

habríamos llegado a un grafo  $G - u$ , conexo con  $\varepsilon(G - u) = \varepsilon(G) - 1$  y  $\nu(G - u) = \nu(G)$ . Utilizando la hipótesis

$$\begin{aligned}\varepsilon(G) &= \nu(G) - 1 \\ \varepsilon(G) - 1 &= \nu(G) - 2 \\ \varepsilon(G - u) &= \nu(G - u) - 2\end{aligned}$$

Esto contradice la hipótesis. Por lo tanto,  $G$  no tiene ciclos, implicando que  $G$  sea un árbol.  $\square$

Por el Teorema 1 para cualquier grafo  $G = (V, E)$  se tiene que  $\sum_{a \in V} gr(a) = 2\varepsilon$ . Pero, por la Proposición 7, si  $G$  es un árbol, entonces  $\varepsilon = \nu - 1$ . Con estos dos resultados, se tendrá que para un árbol,

$$\sum_{a \in V} gr(a) = 2\nu - 2$$

De esta igualdad, podemos deducir el siguiente resultado

**Proposición 8.** *Todo árbol  $G$  con  $\nu \geq 2$  tiene, al menos, dos vértices de grado 1.*

*Demostración.* Supongamos que no hay vértices de grado 1, es decir, que  $gr(a) \geq 2$ , para todo  $a \in V$ . Entonces,

$$2\nu - 2 = \sum_{a \in V} gr(a) \geq \sum_{a \in V} 2 = 2\nu$$

Lo que resulta ser absurdo. Pero tampoco puede ocurrir que haya un sólo vértice  $b$  de grado 1, porque tendríamos

$$2\nu - 2 = \sum_{a \in V} gr(a) = gr(b) + \sum_{a \neq b} gr(a) = 1 + 2(\nu - 1) = 2\nu - 1$$

Así que al menos ha de haber dos vértices de grado 1.  $\square$

## 1.8. Planaridad

Hasta ahora, hemos estudiado muchas propiedades que tienen los grafos sin importar como están dibujados en el plano, ya que esto es solo una construcción auxiliar como se habla en la sección 1.2. Basados en [1], [4] y [6], en esta sección hablaremos de aquellos grafos que se puedan dibujar en el plano sin que ninguna arista se cruce.

**Definición 24.** Una *representación planar* de un grafo  $G$  es una representación gráfica tal que, a los vértices y aristas se le asignan puntos y líneas respectivamente; de manera que ninguna arista se cruce entre sí.

Observemos que una representación planar de un grafo  $G$  se puede ver como un grafo isomorfo a  $G$ . Con esto, daremos paso a la siguiente definición.

**Definición 25.** Sea  $G$  un grafo, se dice *planar* si existe al menos una representación planar de  $G$ .

Es fácil ver que si tomamos la representación planar de cualquier grafo como un grafo, este sería un grafo planar. En la Figura 1.24 vemos un ejemplo de un grafo planar con su representación planar.

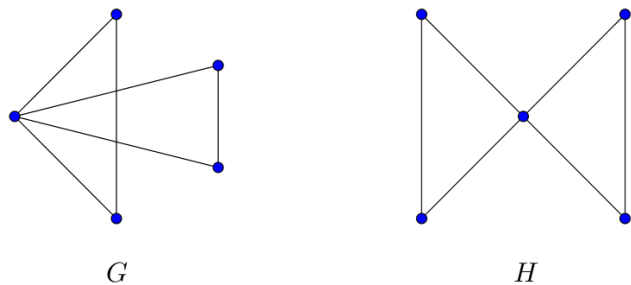


Figura 1.24: El grafo  $H$  es la representación planar del grafo  $G$

Observemos que todo grafo planar divide el plano con sus aristas, lo cual daremos paso a la siguiente definición.

**Definición 26.** Sea  $G$  un grafo plano, una *cara* o *región* es el área del plano que está acotada por aristas. El número de caras que divide un grafo planar  $G$  en el plano lo denotamos como  $\alpha$  o  $\alpha(G)$ . Si  $G$  tiene  $\alpha$  caras, podemos notar cada cara como  $F_1, F_2, \dots, F_\alpha$ ,

Intuitivamente, todo grafo planar tiene exactamente una cara que es infinita, la cual lo llamaremos *cara exterior*. En la Figura 1.25 vemos las caras que tiene el grafo  $G$ , y la cara  $F_4$  es la cara exterior.

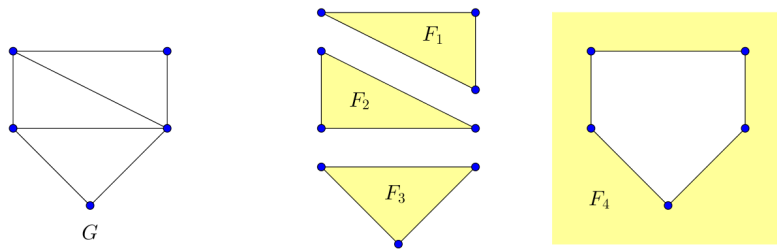


Figura 1.25: Ejemplo de caras

Observemos que cada cara del grafo de la Figura 1.25 está limitada por las aristas del grafo. Esto conlleva a la siguiente definición.

**Definición 27.** Sea  $G$  un grafo planar, el *grado de una cara* es el número de aristas que hay en el borde de una cara. El grado de una cara  $F$  lo denotaremos como  $gr(F)$ .

En la Figura 1.25,  $gr(F_1) = gr(F_2) = gr(F_3) = 3$  y  $gr(F_4) = 5$ . El siguiente teorema relaciona la suma de los grados de las caras con el número de aristas de un grafo planar conexo. Este Teorema es muy parecido al Teorema 1.

**Teorema 3.** *La suma de los grados de las caras de un grafo planar conexo es el doble del número de aristas de dicho grafo. Es decir,*

$$\sum_{i=1}^{\alpha} gr(F_i) = 2\varepsilon$$

*Demostración.* Sea  $G$  un grafo planar conexo, supongamos que  $gr(F_i) = s, i = 1, 2, \dots, \alpha$ . Es decir, hay  $s$  aristas que bordean la cara  $F_i$ , pero las  $s$  aristas también bordean a otras caras  $F_j$  con  $j \neq i$ , por lo que cada arista es contada dos veces y al sumar los grados de las caras, obtenemos el doble del número de las aristas.  $\square$

Leonhard Euler encontró una relación entre el número de vértices, número de aristas, y el número de caras de un grafo plano conexo. Esto se conoce como la *Fórmula de Euler*, la cual mostraremos a continuación.

**Teorema 4.** *Si  $G$  es un grafo planar conexo, entonces*

$$\nu - \varepsilon + \alpha = 2$$

*Demostración.* Utilizaremos el principio de inducción sobre  $\alpha$

Si  $G$  tiene solamente una cara, es decir  $\alpha = 1$ , entonces toda arista de  $G$  es un puente y como  $G$  es conexo, por la Proposición 6 se tiene que  $G$  es un árbol. Así, por la Proposición 7,  $\varepsilon = \nu - 1$  y, por tanto,

$$\nu - \varepsilon + \alpha = \nu - (\nu - 1) + 1 = 2$$

Supongamos ahora que la fórmula es cierta para todos los grafos planares conexos con menos de  $n$  caras. Sea un grafo planar conexo con  $n \geq 2$  caras. Por el hecho de existir más de una cara se puede escoger una arista  $u$  de  $G$  que no sea puente. Entonces  $G - u$  es planar conexo y tiene  $n - 1$  caras, porque las dos caras de  $G$  separadas por  $u$  forman una sola cara en  $G - u$ . Por hipótesis de inducción  $\nu(G - u) - \varepsilon(G - u) + \alpha(G - u) = 2$  y como  $\nu(G - u) = \nu(G)$ ,  $\varepsilon(G - u) = \varepsilon(G) - 1$  y  $\alpha(G - u) = \alpha(G) - 1$  se tiene que

$$\nu(G) - \varepsilon(G) + \alpha(G) = \nu(G - u) - \varepsilon(G - u) - 1 + \alpha(G - u) + 1 = 2$$

En conclusión,  $\nu - \varepsilon + \alpha = 2$  para todo grafo planar conexo.  $\square$

Con el teorema anterior podemos establecer desigualdades que cumplen los grafos planos. Los dos siguientes corolarios mostraran las desigualdades.

**Corolario 5.** *Sea  $G$  un grafo planar simple conexo con  $\nu \geq 3$ . Entonces  $\varepsilon \leq 3\nu - 6$ .*



*Demostración.* Sea  $G$  un grafo planar simple conexo con  $\nu \geq 3$  y  $\alpha$  caras, cada cara tiene al menos tres aristas. En particular, el grado de la cara externa es por lo menos tres. Por lo que  $gr(F_i) \geq 3$  con  $i = 1, 2, \dots, \alpha$ . Ahora,

$$\sum_{i=1}^{\alpha} gr(F_i) \geq \sum_{i=1}^{\alpha} 3 = 3\alpha$$

Aplicando el Teorema 3 tenemos que  $2\varepsilon \geq 3\alpha$ . Por tanto  $\frac{2\varepsilon}{3} \geq \alpha$  y utilizando el Teorema 4, obtenemos

$$\begin{aligned} 2 = \nu - \varepsilon + \alpha &\leq \nu - \varepsilon + \frac{2\varepsilon}{3} \\ 2 &\leq \nu - \frac{\varepsilon}{3} \\ 6 &\leq 3\nu - \varepsilon \\ \varepsilon &\leq 3\nu - 6 \end{aligned}$$

□

**Corolario 6.** Sea  $G$  un grafo planar simple conexo con  $\nu > 3$  y no contiene el ciclo  $\mathcal{C}_3$ , entonces  $\varepsilon \leq 2\nu - 4$

*Demostración.* Sea  $G$  un grafo planar simple conexo con  $\nu > 3$ , supongamos que  $G$  tiene  $\alpha$  caras, como  $G$  no contiene el ciclo  $\mathcal{C}_3$  entonces  $gr(F_i) \geq 4$  con  $i = 1, 2, \dots, \alpha$ . Luego,

$$\sum_{i=1}^{\alpha} gr(F_i) \geq \sum_{i=1}^{\alpha} 4 = 4\alpha$$

Aplicando el Teorema 3 tenemos que  $2\varepsilon \geq 4\alpha$ . Por tanto  $\frac{\varepsilon}{2} \geq \alpha$  y utilizando el Teorema 4, obtenemos

$$\begin{aligned} 2 = \nu - \varepsilon + \alpha &\leq \nu - \varepsilon + \frac{\varepsilon}{2} \\ 2 &\leq \nu - \frac{\varepsilon}{2} \\ 4 &\leq 2\nu - \varepsilon \\ \varepsilon &\leq 2\nu - 4 \end{aligned}$$

□

Como se habló anteriormente, un grafo puede ser planar aunque lo dibujemos con corte entre arista, porque al menos existe una representación planar del grafo. Un ejemplo conocido es el grafo completo  $\mathcal{K}_4$ , generalmente lo dibujamos con corte de aristas pero podemos mover una arista para que no se cruce con ninguna otra, el grafo resultante lo podemos tomar como una representación planar de  $\mathcal{K}_4$ . Podemos ver lo dicho anteriormente con mas claridad en la Figura 1.26

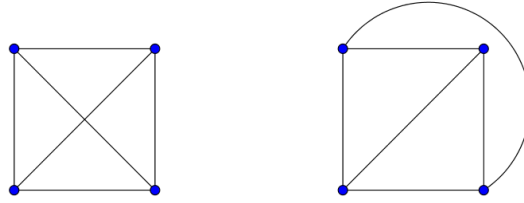
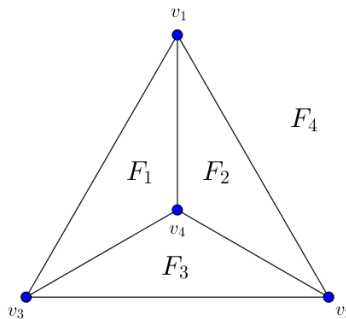


Figura 1.26: Planaridad del grafo completo  $\mathcal{K}_4$

Así como  $\mathcal{K}_4$ , los grafos completos  $\mathcal{K}_1$ ,  $\mathcal{K}_2$  y  $\mathcal{K}_3$  son grafos planares. Pero esto no ocurre con el grafo completo  $\mathcal{K}_5$ , esto lo demostraremos a continuación.

**Proposición 9.** *El grafo completo  $\mathcal{K}_5$  no es planar*

*Demostración.* Llamemos  $v_1, v_2, v_3, v_4, v_5$  a los vértices de  $\mathcal{K}_5$ . Sin pérdida de generalidad, dibujamos los vértices  $v_1, v_2, v_3, v_4$  de manera que ninguna arista se cruce. Dichas aristas forman las caras  $F_1, F_2, F_3, F_4$  como se muestra en la siguiente imagen.

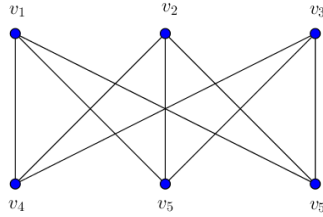


Si  $v_5$  está en  $F_1$  la arista  $\{v_5, v_2\}$  se cruzaría con las aristas que bordean a  $F_1$ , similarmente ocurre si  $v_5$  está en las otras caras. Por lo tanto es imposible dibujar  $\mathcal{K}_5$  sin corte de arista, y así,  $\mathcal{K}_5$  no es planar.  $\square$

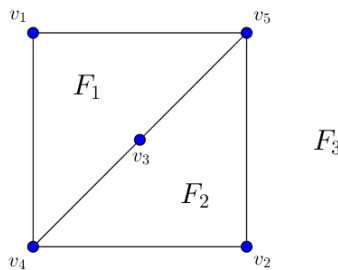
El problema de los servicios públicos trata de que tres casas tengan los tres servicios públicos (agua, electricidad y gas natural) sin que ninguna conexión entre casas y servicios se crucen. Este problema tiene respuesta en la teoría de grafos, que es mostrar si el grafo bipartito completo  $\mathcal{K}_{3,3}$  es planar. Esto se verá en la siguiente proposición, y para demostrarlo utilizamos el mismo método de la Proposición 9.

**Proposición 10.** *El grafo bipartito completo  $\mathcal{K}_{3,3}$  no es planar*

*Demostración.* Sea el grafo bipartito completo  $\mathcal{K}_{3,3}$ . Llamemos  $v_1, v_2, v_3, v_4, v_5, v_6$  a los vértices del grafo, como se muestra en la siguiente imagen.



Sin pérdida de generalidad, dibujamos los vértices  $v_1, v_2, v_4, v_5$  de manera que ninguna arista se cruce. Como  $v_3$  se conecta con  $v_4$  y  $v_5$ , entonces dibujamos esas aristas sin que se crucen con las otras aristas. Dichas aristas forman las caras  $F_1, F_2$  y  $F_3$  como se muestra en la siguiente imagen.



Si  $v_6$  está en  $F_1$  la arista  $\{v_6, v_2\}$  se cruzaría, o con la arista  $\{v_3, v_5\}$  o la arista  $\{v_4, v_3\}$ . Si  $v_6$  está en  $F_2$  la arista  $\{v_6, v_1\}$  se cruzaría, o con la arista  $\{v_3, v_5\}$  o la arista  $\{v_4, v_3\}$ . Si  $v_6$  está en  $F_3$  la arista  $\{v_6, v_3\}$  se cruzaría con cualquiera de las aristas que rodean a  $F_3$ . Por lo tanto es imposible dibujar  $\mathcal{K}_{3,3}$  sin corte de arista, y así,  $\mathcal{K}_{3,3}$  no es planar.  $\square$

Observemos que el grafo completo  $\mathcal{K}_6$  no es planar ya que, al eliminar un vértice con todas las aristas que los conectan, llegamos  $\mathcal{K}_5$ . En general, si un grafo que contiene a  $\mathcal{K}_5$  no es planar; lo mismo ocurre si contiene a  $\mathcal{K}_{3,3}$ . Este hecho es conocido como el *Teorema de Kuratowski*, ya que lo desarrolló el matemático polaco Kazimierz Kuratowski.

**Teorema 7.** *Un grafo no es planar si, y sólo si, contiene un subgrafo que es isomorfo a  $\mathcal{K}_{3,3}$  o  $\mathcal{K}_5$ .*

La demostración de este teorema es muy larga y no va con los objetivos de este trabajo. Esta se encuentra en [6].

## 1.9. Coloración de grafos

En esta sección tomada del texto [3] hablaremos de un caso muy especial del etiquetamiento de grafos, ya que a cada vértice se le asigna un color con ciertas condiciones que se darán en la siguiente definición.

**Definición 28.** Sea  $G = (V, E)$  un grafo y  $C = \{a, b, \dots\}$  un conjunto de colores, una *coloración* de  $G$  con los colores  $C$  consistirá en asignar a cada vértice de  $G$  un elemento de  $C$ , de manera que los extremos de cada arista reciban colores distintos. Formalmente, una coloración es una aplicación

$$\gamma : V \rightarrow C$$

de forma que  $\gamma(v) \neq \gamma(w)$ , si  $\{v, w\} \in E$ . El valor de  $\gamma(v)$  es el color que recibe el vértice  $v$  de la coloración.

Si una coloración utiliza a lo más  $k$  colores, diremos que es una *k-coloración*, y si a un grafo  $G$  se le asigna una *k-coloración*, diremos que  $G$  es *k-coloreable*. Un ejemplo lo tenemos en la Figura 1.27. Ya que podemos colorear los vértices del grafo  $G$  con el conjunto de colores  $C = \{am, az, r, l, v\}$ . Podemos observar que ninguna arista tiene vértices del mismo color. Dicha coloración utiliza 5 colores, por lo que es un 5-coloración y por tanto,  $G$  es 5-coloreable.

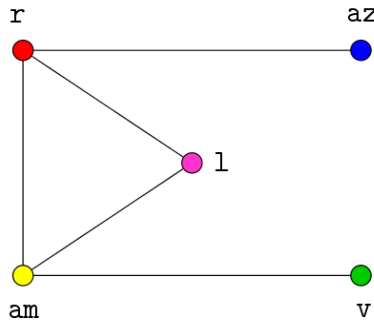


Figura 1.27: Ejemplo de coloración de un grafo

Si un grafo  $G$  es isomorfo a otro grafo  $H$  y  $G$  tiene una coloración, podemos obtener una coloración para  $H$ , de manera que a cada vértice de  $H$  le asignamos el color que lleva el vértice de  $G$  que le corresponde por el isomorfismo. En otras palabras, la coloración es un invariante por isomorfismo.

Aclaremos que en la coloración de grafos se puede restringir a la coloración de grafos simples. Ya que, por ejemplo, si tenemos una coloración de un multigrafo, podemos reducir el multigrafo dejando solamente una aristas en donde existen aristas múltiples, el grafo resultante no cambiará de coloración.

Cualquier grafo  $G$  con  $n$  vértices, podemos asignarle colores distintos a los  $n$  vértices, por lo que  $G$  será  $n$ -coloreable. Si tenemos que  $G$  es  $k$ -coloreable para cualquier  $k \leq n$ , fácilmente  $G$  es  $m$ -coloreable para  $k \leq m \leq n$ . Y si  $G$  no es  $k$ -coloreable entonces no será  $m$ -coloreable si  $m < k$ .

Para colorear un grafo vacío de  $n$  vértices, bastaría utilizar un color, ya que este no posee ninguna arista; y si un grafo es 1-coloreable, no debería tener aristas, ya que si las tuviera, tendríamos vértices adyacente con un mismo color, por lo que el grafo sería un grafo vacío. Así, concluimos que un grafo es 1-coloreable, si y sólo si es un grafo vacío. Mostraremos a continuación, un proposición a cerca de la coloración de grafos bipartitos.

**Proposición 11.** *Un grafo es 2-coloreable si y sólo si es bipartito.*

*Demostración.* Sea  $G$  un grafo 2-coloreable. Sea los colores  $\mathbf{a}$  y  $\mathbf{b}$ , Podemos obtener una partición  $(X, Y)$  del conjunto de vértices tal que  $X$  sea los vértices que tienen asignado el color  $\mathbf{a}$ , y  $Y$  los vértices que tiene asignado el color  $\mathbf{b}$ . Los vértices de  $X$  no pueden estar conectados porque comparten el mismo color, podemos decir lo mismo en conjunto  $Y$ , entonces existen aristas entre un vértice de  $X$  y de  $Y$ . Entonces  $G$  es bipartito.

Recíprocamente, si  $G$  es un grafo bipartito con partición  $(X, Y)$  basta tomar la 2-coloración con colores  $\mathbf{a}$  y  $\mathbf{b}$ , la cual asigna el color  $\mathbf{a}$  a todos los vértices de  $X$  y el color  $\mathbf{b}$  a todos los vértices de  $Y$ . Así,  $G$  es 2-colorable.  $\square$

Si tuviéramos un grafo con un gran número de vértices, no sería muy óptimo utilizar el mismo número de colores para colorear el grafo. Es más interesante utilizar coloraciones con pocos colores, por lo que el número mínimo de colores que se puedan utilizar es una característica muy importante asociada a un grafo. con lo dicho, damos paso a la siguiente definición.

**Definición 29.** El *número cromático* de un grafo  $G$ , denotado por  $\chi$  o por  $\chi(G)$ , será el número mínimo de colores necesario para colorear  $G$ . También decimos que  $G$  es  $\chi$ -cromático

Si un grafo  $G$  es  $\chi$ -cromático, entonces tiene una  $\chi$ -coloración y es  $\chi$ -coloreable. En la Figura 1.27 podemos ver que el grafo tiene una 5-coloración pero en la Figura 1.28 podemos ver el grafo coloreado con 3 colores, lo cual es el número más pequeño para colorear el grafo. Por lo que el número cromático es  $\chi = 3$ .

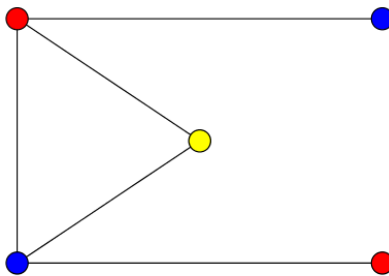


Figura 1.28: El número cromático es  $\chi = 3$

Podemos hallar el número cromático de los tipos de grafos que hablamos en la Sección 1.7. Como el grafo vacío tiene una coloración de un color, entonces el número cromático de este grafo es 1.

En un grafo completo de  $n$  vértices,  $\mathcal{K}_n$ , cada vértice está conectado con los demás vértices, entonces tendríamos que utilizar mínimo  $n$  colores. Por lo tanto,  $\chi(\mathcal{K}_n) = n$ .

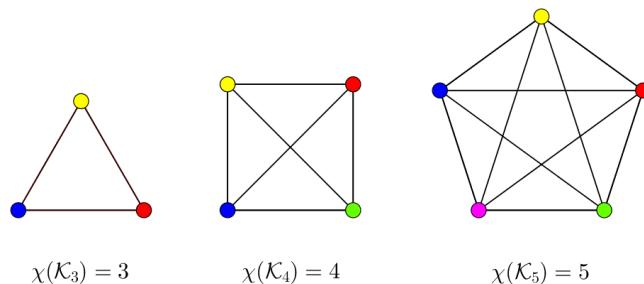


Figura 1.29: Número cromático de los grafos  $\mathcal{K}_3$ ,  $\mathcal{K}_4$  y  $\mathcal{K}_5$

Ya que la Proposición 11 nos dice que cualquier grafo bipartito es 2-coloreable, entonces el número cromático de cualquier grafo bipartito es 2. Entonces, podemos decir también que, los grafos bipartitos completos y los árboles tienen a 2 como número cromático, porque también son bipartitos.

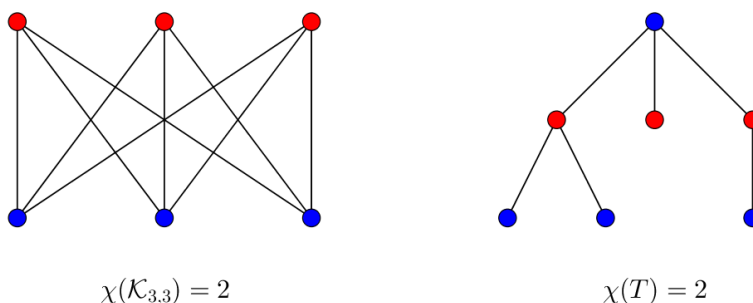


Figura 1.30: Número cromático de  $\mathcal{K}_{3,3}$  y el árbol  $T$

En los ciclos, el número cromático cambia dependiendo del número de vértices.  $\chi(\mathcal{C}_n) = 2$ , si  $n$  es par; esto se da porque los ciclos de un número par de vértices son bipartitos. En los ciclos de número impar de vértices, es imposible utilizar 2 colores; si lo utilizaríamos, el último vértice que colorearíamos sería adyacente al primero que tendría el mismo color, por lo que es necesario, otro color; Luego,  $\chi(\mathcal{C}_n) = 3$ , si  $n$  es impar.

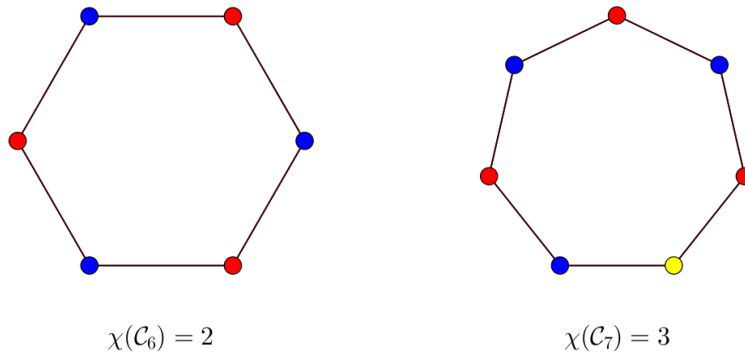


Figura 1.31: Números cromáticos de los ciclos  $\mathcal{C}_6$  y  $\mathcal{C}_7$

A mediados del siglo XIX, Francis Guthrie se dio cuenta mientras coloreaba un mapa de los condados de Inglaterra de que necesitaba al menos cuatro colores para que se cumpliera la condición de que dos regiones con frontera común tuvieran colores distintos (si dos regiones se tocan en un único punto se entiende que no tienen frontera común). Francis le comentó el tema a su hermano Frederick, que a su vez se lo planteó a Augustus de Morgan (profesor suyo en un curso de matemáticas en aquel momento), que aunque no supo responderle se encargó de difundir el asunto entre otros matemáticos. En 1878 Arthur Cayley lo presenta formalmente a la London Mathematical Society y así el problema queda abierto con un enunciado como este: *Todo mapa plano puede colorearse con máximo, cuatro colores con la condición de que regiones con frontera común tengan colores distintos.*

El problema de los cuatro colores afirma que bastan cuatro colores para colorear un mapa geopolítico plano, sin que dos países con frontera común tengan el mismo color. Este problema se puede ver en grafos, asociando una región del mapa y las aristas con las fronteras que tiene las regiones con otras. De esto resulta un grafo plano y conlleva al siguiente teorema.

**Teorema 8.** *Todo grafo planar es 4-coloreable*

En 1879 Alfred Bray Kempe anunció que tenía una demostración para la conjetura. Pero, Percy John Heawood en 1890 encontró un error en la demostración. Heawood no pudo demostrar que la conjetura no era válida, pero probó que con cinco colores se podía colorear cualquier mapa. En 1976 la conjetura tuvo demostración hecha por Kenneth Appel y Wolfgang Haken, en la cual utilizaron un ordenador, esto generó controversias en el ambiente matemático. En la actualidad se realizó otra demostración, también haciendo uso de cálculos por ordenador, lo cual verifica la prueba original; pero sigue sin existir una demostración matemática.

Tomemos como ejemplo el mapa de Sur América y coloreemos con los colores amarillo, azul, rojo y verde.



Figura 1.32: Mapa de Sudamérica

Para colorear este mapa a simple vista es algo que tarda mucho tiempo, para esto se ha creado varias formas de colorear de una forma más fácil y práctica, realizamos un grafo, cada vértice corresponde a un país, a cada vértice se le asigna un color en el cual los vértices vecinos tendrán color diferente. El grafo es planar y también podemos asignarle una 4-coloración con los colores mencionados. Lo cual nos conlleva a la Figura 1.33



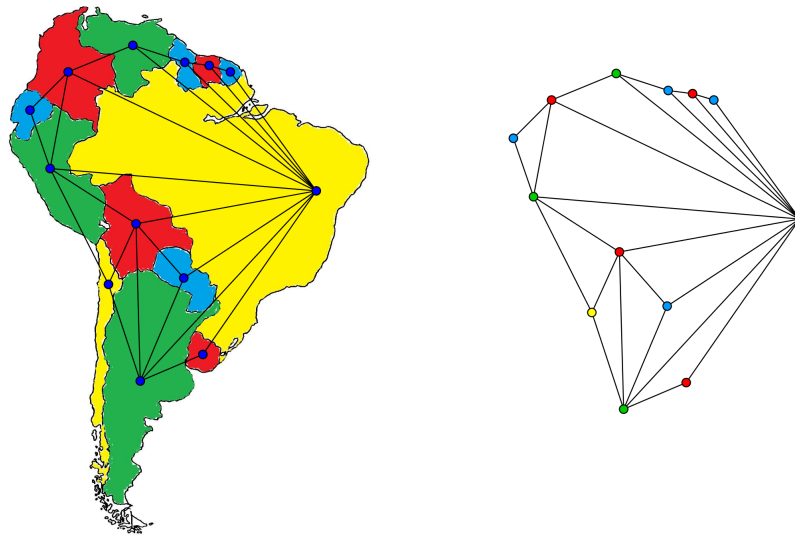


Figura 1.33: Mapa de Sudamérica con su grafo planar 4-coloreable asociado

Ahora tomemos el mapa y formemos un diagrama el cual representaría la distribución de los cuatro colores en los diferentes país también siendo este una aplicación del problema.



Figura 1.34: Distribución de los cuatro colores de Sudamérica

## 1.10. Matriz de adyacencia

En campos como la ciencia y la ingeniería, es conveniente trabajar los grafos como estructura de datos. En [3], [4] y [5] presentamos una forma de representar los grafos como estructura de matriz que es muy usado en programas computacionales y mostraremos sus propiedades. A continuación, daremos su definición.

**Definición 30.** Sea  $G = (V, E)$  un grafo simple con el conjunto de vértices  $V = \{v_1, v_2, \dots, v_n\}$ , su matriz de adyacencia denotada como  $A_G$ , es una matriz de  $n \times n$  donde el elemento de la matriz  $a_{ij}$  es 1 cuando hay una arista que une los vértices  $v_i$  y  $v_j$ ; en caso contrario el elemento  $a_{ij}$  es 0, para  $1 \leq i, j \leq n$ . En términos de la relación la matriz de adyacencia se puede definir como  $A_G \in M_{n \times n}(\mathbb{Z}_2)$  mediante

$$A_{G_{ij}} = \begin{cases} 1 & \text{si } v_i \text{ r } v_j, \\ 0 & \text{en caso contrario} \end{cases}$$

Además de ser simétrica por  $a$  donde determina que el grafo es simple.

	$v_1$	$v_2$	$v_3$	$\dots$	$v_n$
$v_1$	0	1	0	$\dots$	1
$v_2$	1	0	1	$\dots$	1
$v_3$	0	1	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$v_n$	1	1	0	$\dots$	0

Si el grafo es simple su matriz de adyacencia tendrá ceros en la diagonal y será simétrica, esto cambia si el grafo es un multigrafo o digrafo. El siguiente teorema relaciona la matriz de adyacencia de un grafo con el número de paseos de longitud  $k$  que tiene dicho grafo.

**Teorema 9.** Si  $M$  es la matriz de adyacencia de un grafo  $G$ , la entrada  $(i, j)$  de la matriz  $M^k$  cuenta el número de paseos de longitud  $k$  entre los vértices  $v_i$  y  $v_j$ .

*Demostración.* Lo hacemos por inducción sobre  $k$ . El caso  $k = 1$  será la matriz de adyacencia. Llamemos  $a_{ij}^{(k)}$  a la entrada que ocupa la posición  $(i, j)$  en la matriz  $M^k$ . Como  $M^k = M^{k-1}M$  las reglas de multiplicación de matrices nos dicen que

$$a_{ij}^{(k)} = \sum_{l=1}^n a_{il}^{(k-1)} a_{lj}.$$

Observemos ahora que  $a_{il}^{(k-1)} a_{lj} = a_{il}^{(k-1)}$  si es que  $v_l$  es vecino de  $v_j$ , mientras que vale 0 en caso contrario. Por hipótesis,  $a_{il}^{(k-1)}$  cuenta el número de paseos de longitud  $k - 1$  entre  $v_i$  y  $v_l$ . Así que, en la suma de arriba, estamos contando todos los posibles paseos de  $v_i$  a  $v_j$  de longitud  $k$  clasificándolos en función del vértice que, en cada paseo, sea justo el anterior a  $v_j$ .  $\square$

**Ejemplo 3.** Tomemos el grafo de la Figura 1.35.

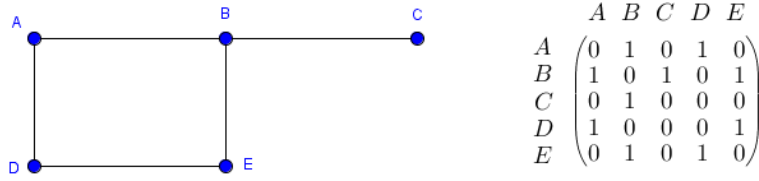


Figura 1.35: Grafo con su respectiva matriz de adyacencia

La matriz que esta a la derecha del grafo relaciona la conexión que tiene todos los vértices. Entonces, la matriz de adyacencia  $M$  sería la siguiente:

$$M = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix},$$

Ahora, para aplicar el teorema anterior, elevemos la matriz a las potencias 2 y 3. Las matrices resultantes son las siguientes.

$$M^2 = \begin{bmatrix} 2 & 0 & 1 & 0 & 2 \\ 0 & 3 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 2 & 0 \\ 2 & 0 & 1 & 0 & 2 \end{bmatrix} \quad M^3 = \begin{bmatrix} 0 & 5 & 0 & 4 & 0 \\ 5 & 0 & 3 & 0 & 5 \\ 0 & 3 & 0 & 2 & 0 \\ 4 & 0 & 2 & 0 & 4 \\ 0 & 5 & 0 & 4 & 0 \end{bmatrix}$$

Observemos que en la diagonal de  $M^2$  aparecen los grados de cada vértice. Esto es en general, la entrada  $a_{ii}^{(2)}$  coincide con el grado del vértice  $v_i$ . De la matriz  $M^3$  deducimos, por ejemplo, que hay cinco paseos distintos de longitud tres entre los vértices  $A$  y  $B$ . Los cinco paseos entre  $A$  y  $B$  en el grafo son  $A - D - A - B$ ,  $A - B - C - B$ ,  $A - B - E - B$ ,  $A - D - E - B$  y  $A - B - A - B$ .

Sea  $G$  un grafo con  $n$  vértices y su respectiva matriz de adyacencia  $M$ . Observemos que, si dos vértices se pueden conectar en un grafo, es seguro que lo podrán hacer utilizando un camino simple de longitud a lo sumo  $n - 1$ . La entrada  $(i, j)$  de la matriz

$$\widetilde{M} = I + M + M^2 + \dots + M^{(n-1)}$$

contiene la información sobre el número de paseos, de longitud a lo sumo  $n - 1$  que existen entre los vértices  $v_i$  y  $v_j$ . Por lo tanto, si alguna entrada de la matriz  $\widetilde{M}$  es nula, entonces el grafo no puede ser conexo y viceversa. Si todas las entradas de  $\widetilde{M}$  son positivas entonces el grafo será conexo. Si el grafo no es conexo, la matriz  $\widetilde{M}$  también contiene la información sobre las componentes conexas. Tomemos por ejemplo, la primera fila de la matriz, etiquetada con el vértice  $v_1$ , las posiciones en las que

aparezcan unos determinan los vértices de la componente conexas a la que pertenece  $v_1$ . El procedimiento se repetirá para la fila correspondiente al primer vértice no incluido en la componente anterior. Y así sucesivamente, hasta determinar todas las componentes conexas del grafo.

**Teorema 10.** *Dos grafos  $G = (V, E)$  y  $H = (U, F)$  con matrices de adyacencia  $A_G$  y  $A_H$  respectivamente son isomorfos si, y sólo si, existe una matriz permutan  $P$  tal que  $A_G = PA_HP'$  o sea que  $A_G$  y  $A_H$  son semejantes.*

*Demostración.* Si  $G$  y  $H$  son isomorfos, entonces  $|V| = |U| = n$  para cierto  $n \in \mathbb{N}$ , sean  $A_G, A_H \in M_{n \times n}(\mathbb{Z}_2)$  las matrices de adyacencia de  $G$  y  $H$ , relativas a cierta enumeración  $\bar{v} = (v_1, v_2, \dots, v_n)$  de  $V$  y  $\bar{u} = (u_1, u_2, \dots, u_n)$  de  $U$  donde  $\bar{v}, \bar{u}$  se puede identificar como elementos de  $\mathbb{R}^n$ , el isomorfismo se expresa por una matriz de permutación (ortogonal)  $P \in M_{n \times n}(\mathbb{Z}_2)$  donde  $\bar{u} = P\bar{v}$ . Consideremos la forma que toma el endomorfismo lineal  $\bar{y} = A_G\bar{x}$  de  $\mathbb{R}^n$ , cuando ocurre el cambio de base  $\bar{x} = P\tilde{x}$ ,  $\bar{y} = P\tilde{y}$ . Luego se tiene  $P\tilde{y} = (A_GP)\tilde{x}$ , o sea  $\tilde{y} = (P^tA_GP)\tilde{x}$ . Así,  $A_H = P^tA_GP$ .

Recíprocamente, si  $A_H = P^tA_GP$ , para cierta matriz de permutación  $P \in M_{n \times n}(\mathbb{Z}_2)$ , con  $A_G, A_H \in M_{n \times n}(\mathbb{Z}_n)$  para cierto  $n \in \mathbb{N}$  luego  $|V| = |U| = n$ , ahora enumerando  $V$  como  $v_1, v_2, \dots, v_n$  y  $U$  como  $u_1, u_2, \dots, u_n$ , para obtener las matrices de adyacencia respectivas mencionadas  $A_G$  y  $A_H$ , se define la biyección  $\phi : V \rightarrow U$  mediante  $\phi(v_i) = u_j$ , si  $P_{ij} = 1$ . Sabemos que  $v_i$  a  $v_j$  en  $V$  si y sólo si  $A_{G_{ij}} = 1$ . La Matriz  $A_GP$  es una permutación de la columnas de  $A_G$  y la matriz  $A_H = P^tA_GP$  es la permutación inversa ( $P^t = P^{-1}$ ) de las filas de  $A_GP$ . Entonces  $A_{H_{kl}} = 1$  si y sólo si  $u_k = \phi(v_i)$  y  $u_l = \phi(v_j)$ , es decir  $\phi(v_i)$  a  $\phi(v_j)$  en  $U$  si y sólo si  $v_i$  a  $v_j$  en  $V$ . Se concluye que  $\phi$  es un isomorfismo de grafos.  $\square$

Tomemos como ejemplo los grafos  $G$  y  $H$  que muestra la Figura 1.36

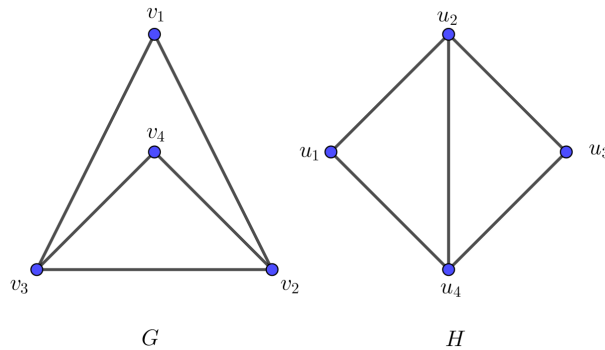


Figura 1.36:  $G$  y  $H$  son isomorfos

La matriz de adyacencia de los grafos son:

$$A_G = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad A_H = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Observemos la biyección de los vértices del grafo  $G = (V, E)$  a el grafo  $H = (U, F)$

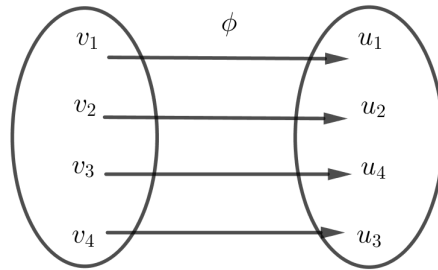


Figura 1.37

La siguiente es la matriz de permutación  $P$ .

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_4 \\ u_3 \end{pmatrix}$$

Tomamos la matriz inversa de  $P$ , pero sabemos que el elemento inverso de la matriz de permutación es correspondiente a la matriz transpuesta.

$$P^{-1} = P^t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Ahora tomemos  $A_G$

$$A_G = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$P^t A_G P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Realizamos multiplicación de matrices.

Multiplicamos  $P^t A_G$

$$P^t A_G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

El resultado obtenido anteriormente se multiplicará con la matriz  $P$

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Podemos observar que el último resultado corresponde a la matriz de adyacencia de  $A_H$ . Luego  $P^t A_G P = A_H$ .

# Capítulo 2

## Algoritmo de grafos

En este capítulo enfatizaremos en los algoritmos de grafos, especialmente en el algoritmo de la matriz de adyacencia y los clásicos algoritmos de recorrido de un grafo, la búsqueda de profundidad y búsqueda de anchura. Describiremos cada algoritmo, realizaremos ejemplos y mostraremos su implementación en el programa Mathematica. No sin antes, dar las bases del entorno de dicho programa. Cabe recalcar que los algoritmos de recorrido sobre grafos se realizará solamente en grafos simples para este trabajo.

La palabra algoritmo viene del griego y latín, *dixit algorithmus* y este a su vez del matemático persa Al-Juarismi. En la vida cotidiana, los algoritmos se usan muy frecuentemente, como lo son los manuales de usuario, las instrucciones que recibe un trabajador por parte de su superior, las recetas de cocina, e incluso, el quehacer de una persona en un día. En las matemáticas, está el algoritmo de la multiplicación y la división, que se usan para hallar el producto y el cociente de dos números respectivamente, el algoritmo de Euclides con el cual se obtiene el máximo común divisor de dos enteros, y el método de Gauss, que se usa para resolver un sistema de ecuaciones lineales, entre otros.

### 2.1. Mathematica

*Mathematica* es un programa matemático de computación simbólica, que es muy utilizado en campos científicos, de ingeniería, matemáticas y computación. Constituye una herramienta muy potente para la realización de todo tipo de cálculos matemáticos, desde los más básicos hasta los más complejos. Las últimas versiones del programa incorporan un editor de texto y permiten el manejo de paletas de símbolos que facilitan el uso de los comandos e instrucciones del programa y la utilización de la simbología habitual en las fórmulas y operaciones matemáticas. El programa Mathematica está estructurado en 2 partes: El *Front End* y el *Kernel* o *Núcleo*.

En Mathematica, los documentos tienen la extensión “.nb” y se denominan *Notebooks*, la cual permiten añadir texto, gráficos, sonidos y animaciones, ofreciendo la posibilidad de ejecutar el documento de forma interactiva.

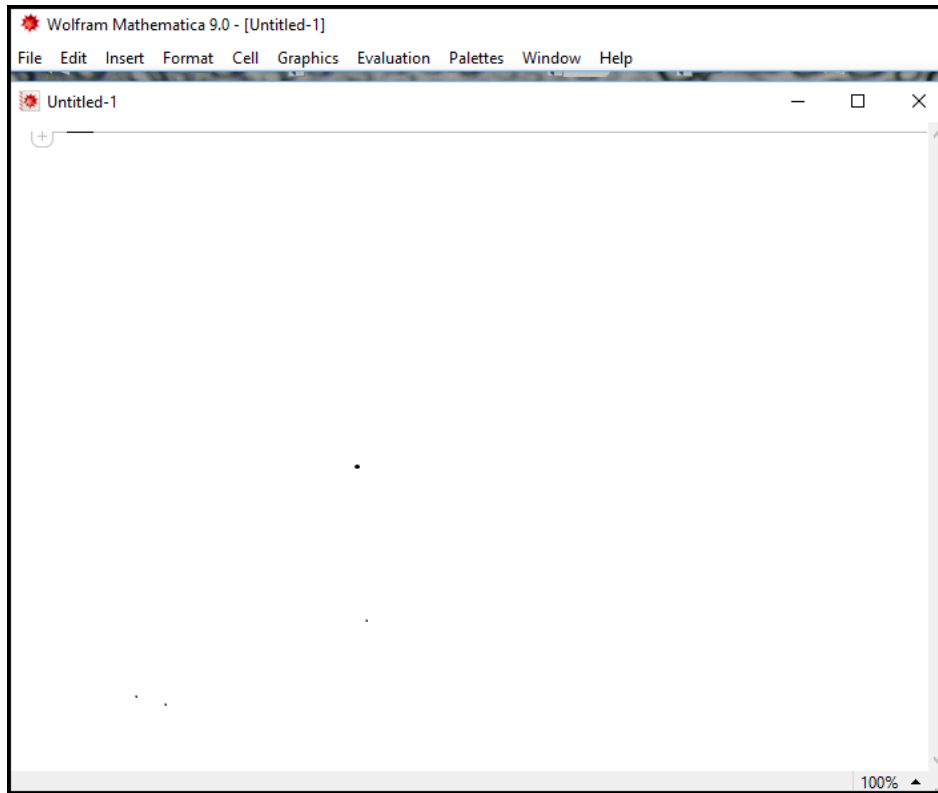


Figura 2.1: Notebook de Mathematica

A continuación daremos algunas orientaciones básicas en el manejo de grafos en Mathematica. Para definir un grafo en Mathematica, utilizamos el comando

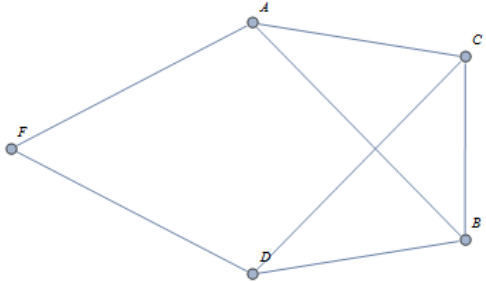
$$\text{Graph}[\{v_1, v_2, \dots\}, \{e_1, e_2, \dots\}]$$

donde cada  $v_i$  son los vértices del grafo, que pueden ser definidos con cualquier letra o número; y cada  $e_i$  son las aristas que conecta dichos vértices. Si una arista es dirigida, se define con el comando  $v_i \rightarrow v_j$  o con `DirectedEdge[vi, vj]`, y si una arista es no dirigida, se define con  $v_i \leftrightarrow v_j$  o con `UndirectedEdge[vi, vj]`. Si queremos que en el grafo los vértices sean nombrados, al comando `Graph` le agregamos `VertexLabels→"Name"`. A continuación, daremos una lista de comandos que nos darán mas información del grafo y en la Figura 2.2 aplicamos estos comandos.

- `VertexList[G]`: Muestra la lista de vértices del grafo  $G$
- `EdgeList[G]`: Muestra la lista de aristas del grafo  $G$
- `VertexCount[G]`: Muestra el número de vértices del grafo  $G$
- `EdgeCount[G]`: Muestra el número de aristas del grafo  $G$



```

In[1]:= G = Graph[{A, B, C, D, F}, {A <-> B, C <-> D, F <-> A, C <-> A, D <-> F,
                B <-> C, D <-> B}, VertexLabels -> "Name"]
Out[1]= 

```

```

In[2]:= VertexList[G]
Out[2]= {A, B, C, D, F}

In[3]:= EdgeList[G]
Out[3]= {A -> B, C -> D, F -> A, C -> A, D -> F, B -> C, D -> B}

In[4]:= VertexCount[G]
Out[4]= 5

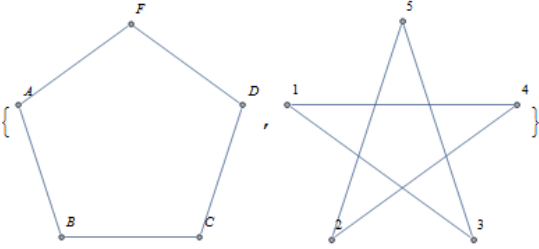
In[5]:= EdgeCount[G]
Out[5]= 7

```

Figura 2.2: Aplicación del comando `Graph` en Mathematica

En Mathematica, podemos mirar si un grafo es conexo, planar, e isomorfo a otro grafo. Para esto, utilizamos los comandos `ConnectedGraphQ[G]`, `PlanarGraphQ[G]` y `IsomorphicGraphQ[G,H]`. En la Figura 2.3 mostraremos una aplicación de estos comandos.

```

In[1]:= {G, H} =
  {Graph[{A, B, C, D, F}, {A <-> B, B <-> C, C <-> D, D <-> F, F <-> A},
    VertexLabels -> "Name"],
  H = Graph[{1, 2, 3, 4, 5}, {1 <-> 4, 4 <-> 2, 2 <-> 5, 5 <-> 3, 3 <-> 1},
    VertexLabels -> "Name"]}
Out[1]= 

```

```

In[2]:= IsomorphicGraphQ[G, H]
Out[2]= True

In[3]:= ConnectedGraphQ[G]
Out[3]= True

In[4]:= PlanarGraphQ[G]
Out[4]= True

```

Figura 2.3: Mas comandos de grafos en Mathematica

Ahora podemos proseguir a estudiar los algoritmos de grafos que se mencionaron anteriormente. Todos los algoritmos que involucran grafos nos ayudan a resolver los problemas que tenemos con los grafos, como lo es la planaridad y la conexidad. Los que vamos a estudiar especialmente, la búsqueda de profundidad y de anchura, estos algoritmos tienen como fin el recorrido que se puede realizar un grafo. Para ello, visitan un vértice del grafo y dependiendo del recorrido del que se requiera, visitamos a los otros nodos conectados a este. Pero, primero comenzaremos con el algoritmo de la matriz de adyacencia; como indica su nombre, halla la matriz de adyacencia de un grafo. Para tener más información y detalles del software Mathematica se encontrará en el texto [9]

## 2.2. Algoritmo de la matriz de adyacencia

Como se habló en el Capítulo anterior, la matriz de adyacencia es una matriz en la cual se puede representar un grafo por medio de los vértices adyacentes que tiene dicho grafo. El algoritmo consiste en escribir 1 en la posición  $ij$  de la matriz si los vértices  $i$  y  $j$  son adyacentes; en caso contrario la posición  $ij$  es 0. En la Figura 2.4, mostraremos el diagrama de flujo.

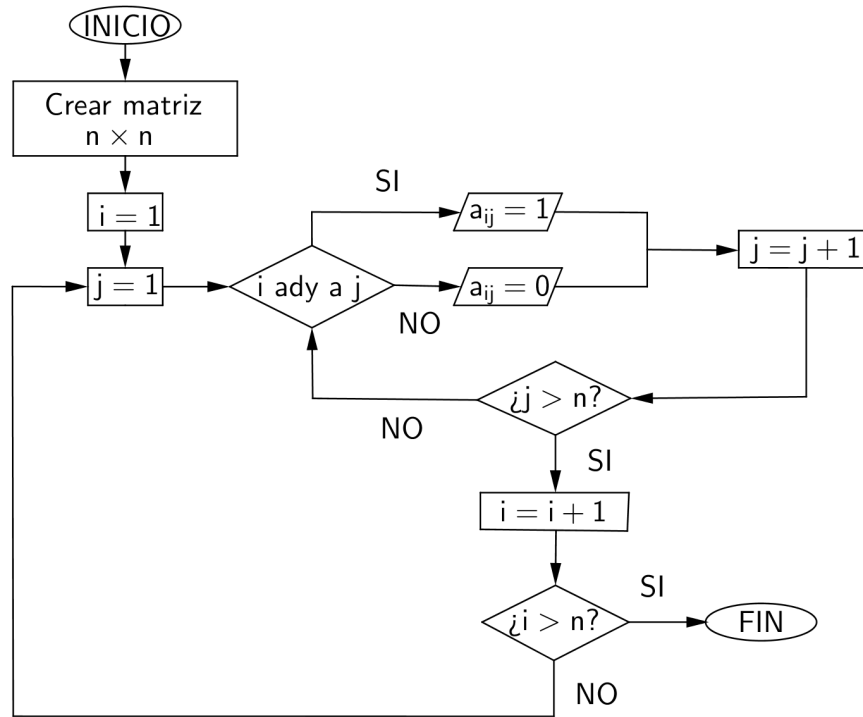


Figura 2.4: Diagrama de flujo del algoritmo de la matriz de adyacencia

Ahora realizaremos la prueba de escritorio para este algoritmo. Sea  $G$  un grafo con vértices  $V_G = \{1, 2, 3, 4\}$  y aristas dirigidas  $E_G = \{\{1, 3\}, \{1, 2\}, \{1, 4\}, \{3, 2\}, \{2, 4\}, \{4, 1\}\}$ . Hallemos la matriz de adyacencia de este grafo.

- Sea  $i = 1$  y  $j = 1$ .  $i$  y  $j$  no son adyacentes. Entonces, marcamos 0 en la fila uno columna uno.

	1	2	3	4
1	<b>0</b>	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

- Sea  $i = 1$ ,  $j = 2$  y  $j \leq 4$ .  $i$  y  $j$  son adyacentes. Entonces marcamos 1 en la fila uno columna dos.

	1	2	3	4
1	<b>0</b>	<b>1</b>	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

- Sea  $i = 1$ ,  $j = 3$  y  $j \leq 4$ .  $i$  y  $j$  son adyacentes. Entonces, marcamos 1 en la fila uno columna tres.

	1	2	3	4
1	<b>0</b>	<b>1</b>	<b>1</b>	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

- Sea  $i = 1$ ,  $j = 4$  y  $j \leq 4$ .  $i$  y  $j$  son adyacentes. Entonces, marcamos 1 en la fila uno columna cuatro.

	1	2	3	4
1	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

Realizamos este mismo procedimiento cuando  $i = 2, 3, 4$ . Como resultado, obtenemos la siguiente matriz de adyacencia.

	1	2	3	4
1	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>
2	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>
3	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>
4	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>

En la Figura 2.5 vemos el resultado en Mathematica del algoritmo de la matriz de adyacencia.

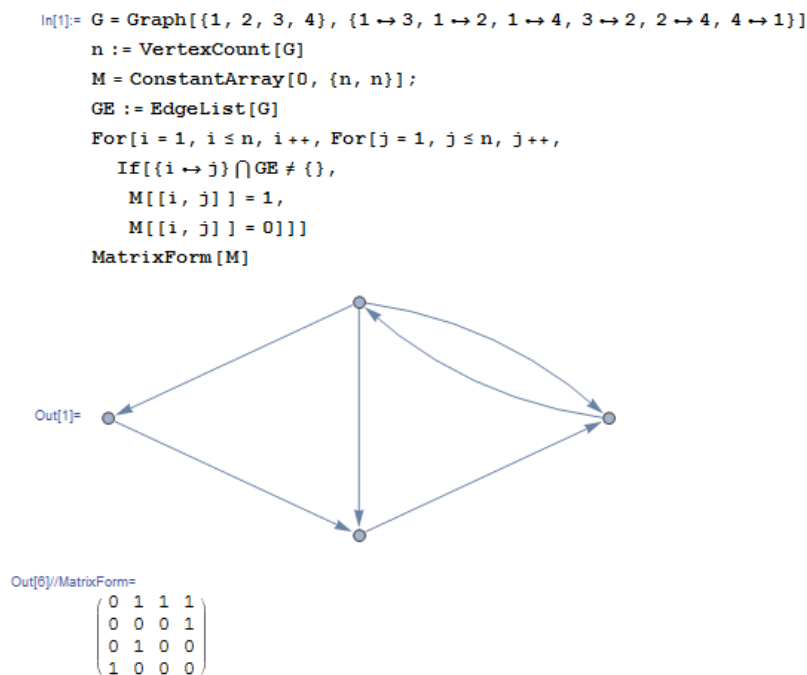


Figura 2.5: Programa de la matriz de adyacencia en Mathematica

### 2.3. Algoritmo de búsqueda de profundidad (DFS)

Según el texto [5], la búsqueda de profundidad, *Depth First Search* en inglés y abreviado DFS, es un algoritmo que permite recorrer todos los nodos de un grafo de manera ordenada y no uniforme. La estrategia que sigue este algoritmo es, como su nombre indica, buscar “más profundamente” en un grafo siempre que sea posible. Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando, de forma recurrente, en un camino concreto. Cuando ya no quedan más nodos que visitar en dicho camino, regresa, de modo que repite el mismo proceso con cada uno de los nodos ya procesados. Este algoritmo utiliza una lista de espera la cual llamaremos STACK, está lista almacena aquellos vértices que son visitados y se eliminan cuando estos vértices ya no tienen adyacencia de vértices no visitados. Al finalizar el algoritmo, el STACK debe estar vacío.

El algoritmo funciona dando un vértice inicial  $A$ , el cual lo marcamos como visitado. Luego marcamos como visitado cada vértice  $B$  a lo largo de un camino  $P$  que empieza en  $A$ ; es decir, se marca como visitado vecino de  $A$ , luego un vecino del vecino de  $A$  y así sucesivamente. al llegar a un vértice sin vecino no visitado, se retrocede en el camino  $P$  hasta que se encuentre un vértice en el que tenga vecinos no visitados y se pueda seguir por otro camino  $P'$ . Y así en lo sucesivo. El retroceso se logra usando el STACK para mantener los vértices iniciales de posibles caminos futuros. A continuación

observemos de diagrama de flujo para el algoritmo de búsqueda de profundidad en la Figura 2.6.

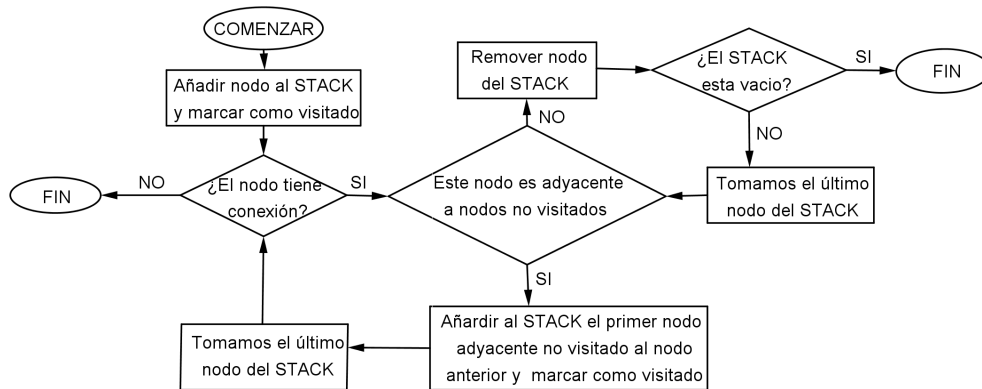


Figura 2.6: Diagrama de flujo del DFS

Ahora, realizaremos una prueba de escritorio tomando el grafo de la Figura 2.7. Tomamos como nodo inicia el nodo *A*.

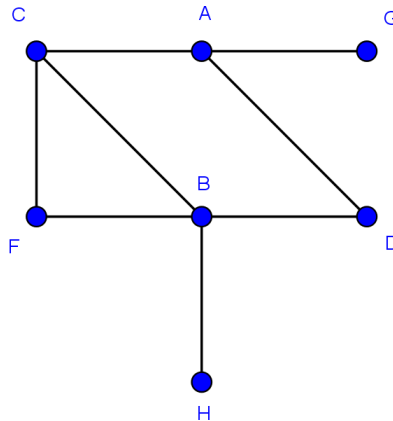


Figura 2.7: Grafo de las pruebas de escritorio del DFS y BFS

El nodo *A* lo agregamos al STACK y lo marcamos como visitado. El nodo *A* tiene nodos adyacentes que no son visitados y son los nodos *C*, *G* y *D*. Alfabéticamente elegimos el nodo *C* el cual agregamos al STACK y lo marcamos como visitado; por el momento tendríamos lo siguiente:

- STACK: *A*, *C*
- DFS: *A*, *C*

El nodo *C* tiene nodos adyacentes no visitados las cuales son los nodos *B* y *F*. Tomamos el nodo *B*, lo marcamos como visitados y lo agregamos al STACK. por lo que tendríamos lo siguiente:

- STACK:  $A, C, B$
- DFS:  $A, C, B$

El nodo  $B$  tiene como nodos adyacentes los nodos  $D, F$ , y  $H$ . Tomamos el nodo  $D$ , lo marcamos como visitados y lo agregamos al STACK.

- STACK:  $A, C, B, D$
- DFS:  $A, C, B, D$

El nodo  $D$  no tiene nodos adyacentes no visitados, por lo que lo removemos del STACK, y como el STACK no está vacío, tomamos el último nodo de la lista, que sería el nodo  $B$  y miramos el siguiente nodo no visitado adyacente a  $B$  que es el nodo  $F$ , lo agregamos al STACK y lo marcamos como visitado. Así tendríamos:

- STACK:  $A, C, B, F$
- DFS:  $A, C, B, D, F$

El nodo  $F$  no tiene nodos adyacentes no visitados, por lo que lo removemos del STACK, y como el STACK no está vacío, tomamos el último nodo de la lista, que sería el nodo  $B$  y miramos el último nodo no visitado adyacente a  $B$  que es el nodo  $H$ , lo agregamos al STACK y lo marcamos como visitado.

- STACK:  $A, C, B, H$
- DFS:  $A, C, B, D, F, H$

El nodo  $H$  no tiene nodos adyacentes no visitados, por lo que lo removemos del STACK, y como el STACK no está vacío, tomamos el último nodo de la lista, que sería el nodo  $B$ .  $B$  ya no tiene nodos adyacentes no visitados. por lo que lo removemos del STACK y volvemos al nodo  $C$ .

- STACK:  $A, C$
- DFS:  $A, C, B, D, F, H$

El nodo  $C$  ya no tiene nodos no visitados, lo removemos del STACK y volvemos al nodo  $A$ . El nodo  $A$  ahora tiene un nodo no visitado que es el nodo  $G$ , por lo que lo agregamos al STACK y lo marcamos como visitado.

- STACK:  $A, G$
- DFS:  $A, C, B, D, F, H, G$

El nodo  $G$  no tiene nodos adyacentes, por lo que lo removemos del STACK, y volvemos al nodo  $A$ . El nodo  $A$  ya no tiene nodos adyacentes no visitados, por lo que lo removemos del STACK, y como el STACK ya está vacío, se termina la búsqueda. Y el resultado es:

- STACK:

- DFS:  $A, C, B, D, F, H, G$

Y así este termina la búsqueda de profundidad del grafo y el recorrido es  $A, C, B, D, F, H, G$ .

En Mathematica, podemos realizar la búsqueda de profundidad de un grafo dirigido y no dirigido, esto lo podemos lograr con el comando:

```
DepthFirstScan[g, v, {"PrevisitVertex"→(Print[#]&)}]
```

Donde  $g$  es el grafo,  $v$  es el vértice del grafo donde comienza la búsqueda. Este nos arroja una lista de los vértice hecho por la búsqueda. En la Figura 2.8 aplicamos el comando con el grafo de la Figura 2.7.

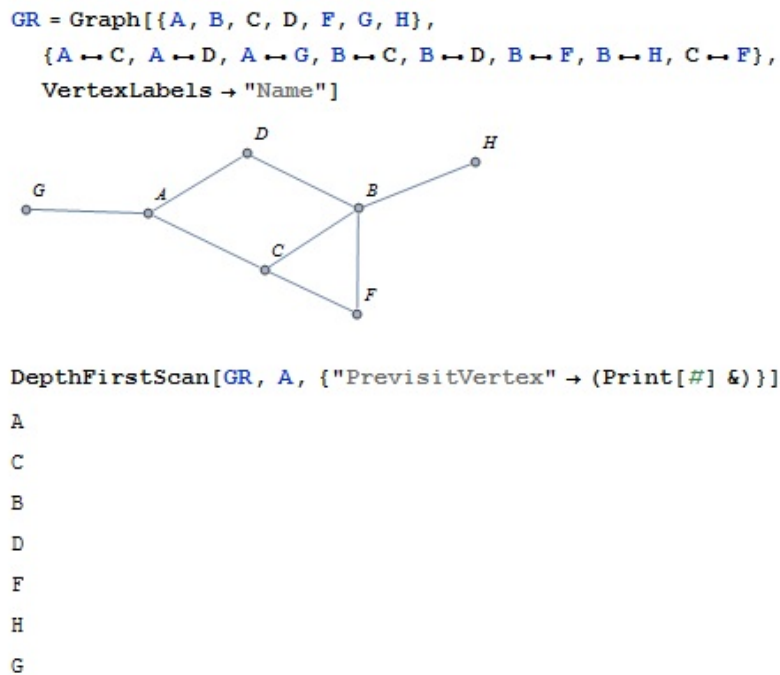


Figura 2.8: Comando de las búsqueda de profundidad

Podemos observar que al hacerle una búsqueda de profundidad a un grafo  $G$ , podemos obtener un subárbol que es un subgrafo extendido de  $G$ . En Mathematica, con el comando:

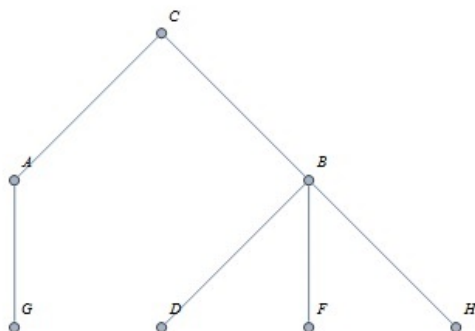
```
T=Reap[DepthFirstScan[g, v, {"FrontierEdge"→Sow}]] [[2, 1]]
Graph[T, VertexLabels→"Name"]
```

nos muestra dicho subarbol. Y el comando:

```
Reap[DepthFirstScan[g, v, {"FrontierEdge"→Sow}]] [[2, 1]]
//HighlightGraph[g,#]&
```

refleja el subárbol en el grafo original. En la Figura 2.9 nos muestra la aplicación de estos comandos.

```
T = Reap[DepthFirstScan[GR, A, {"FrontierEdge" → Sow}]][[2, 1]]
Graph[T, VertexLabels → "Name"]
{A → C, C → B, B → D, B → F, B → H, A → G}
```



```
Reap[DepthFirstScan[GR, A, {"FrontierEdge" → Sow}]][[2, 1]]
// HighlightGraph[GR, #] &
```

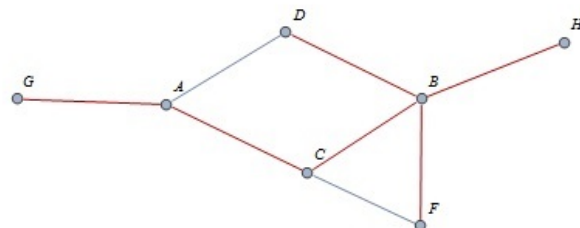


Figura 2.9: Más comandos del algoritmo de profundidad

## 2.4. Algoritmo de búsqueda de anchura (BFS)

Como se aprecia en el texto [5], la búsqueda de anchura, en ingles *Breadth First Search* y abreviado BFS, es un algoritmo para recorrer o buscar elementos en un grafo. Este algoritmo es usado frecuentemente sobre árboles. Intuitivamente, se comienza en la raíz, que puede ser un nodo cualquiera de un grafo, y se exploran todos los vecinos de este nodo. A continuación para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo los vértices de un grafo. Este algoritmo, así como el anterior, utiliza una lista de apilamiento, la cual la llamamos QUEUE, este almacena todos los vértices adyacentes al vértice visitado mientras elimina a este último.

La búsqueda en anchura empieza en un vértice inicial A que el cual lo marcamos como visitado. Luego se marcan como visitados todos los vecinos de A. Enseguida se marcan como visitados todos los vecinos de los vecinos de A. Lo natural es seguir la pista de los vecinos de un vértice, y es necesario garantizar que ningún vértice sea procesado dos veces. Para esto utilizamos el QUEUE para mantener los vértices que



están en espera de ser marcados como visitados. Veremos el diagrama de flujo de este algoritmo en la Figura 2.10.

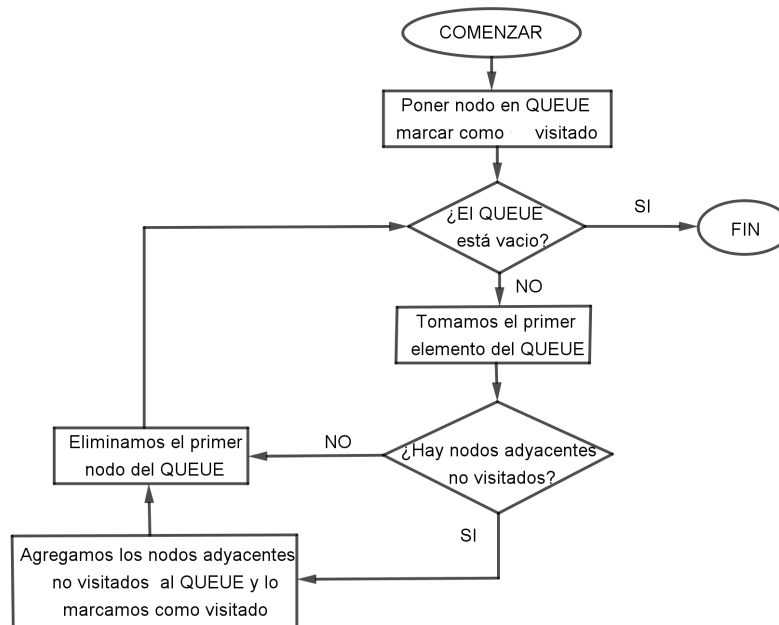


Figura 2.10: Diagrama de flujo del algoritmo de búsqueda de anchura

Ahora, realizaremos la prueba de escritorio con el grafo de la Figura 2.7 y nuestro nodo inicial al nodo *A*.

El nodo *A* lo agregamos al QUEUE y lo marcamos como visitado. lo cual tendremos por el momento lo siguiente

- QUEUE: *A*
- BFS: *A*

El nodo *A* tiene nodos adyacentes que no son visitados y son los nodos *C*, *D* y *G*. Estos nodos los agregamos al QUEUE y lo marcamos como visitado y eliminados del QUEUE al vértice *A*. Así tendríamos lo siguiente

- QUEUE: *C*, *D*, *G*
- BFS: *A*, *C*, *D*, *G*

el primer nodo que encabeza el QUEUE es el nodo *C*, por lo que miramos cuales nodos adyacentes a *C* son no visitados, esos nodos son el *B* y *F*. por lo que lo agregamos al QUEUE, los marcamos como visitados y eliminamos al vértice *C*. Así tendríamos:

- QUEUE: *D*, *G*, *B*, *F*
- BFS: *A*, *C*, *D*, *G*, *B*, *F*

Ahora, el primer nodo que encabeza el QUEUE es el nodo  $D$ . Pero, este nodo no tiene nodos adyacentes no visitados, por lo que lo eliminamos del QUEUE y seguimos con el nodo  $G$ . Así como el nodo  $C$ , el nodo  $G$  no tiene nodos adyacentes no visitados, por lo que lo eliminamos del QUEUE. Así, tendríamos lo siguiente:

- QUEUE:  $B, F$
- BFS:  $A, C, D, G, B, F$

Ahora, el nodo  $B$  encabeza el QUEUE y tiene como nodo no visitado el nodo  $H$ , lo marcamos como visitado, lo agregamos al QUEUE y eliminamos al nodo  $B$ . Así, tendríamos:

- QUEUE:  $F, H$
- BFS:  $A, C, D, G, B, F, H$

Ahora, observemos el nodo  $F$ , pero este no tiene nodos adyacentes no visitados, por lo que lo eliminamos del QUEUE. Lo mismo pasa con el nodo  $H$ . Finalmente, tenemos lo siguiente

- QUEUE:
- BFS:  $A, C, D, G, B, F, H$

Y así este termina la búsqueda de anchura del grafo y el recorrido es  $A, C, D, G, B, F, H$

Así como en el algoritmo de búsqueda de profundidad, en Mathematica, podemos realizar la búsqueda de anchura de un grafo dirigido y no dirigido, esto lo podemos lograr con el comando:

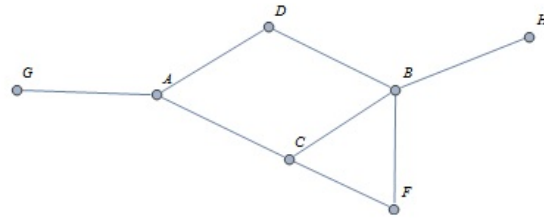
```
BreadthFirstScan[g, v, {"PrevisitVertex"→(Print[#]&)}]
```

Donde  $g$  es el grafo,  $v$  es el vértice del grafo donde comienza la búsqueda. En la Figura 2.11 aplicamos el comando con el grafo de la Figura 2.7.

```

In[15]:= GR = Graph[{A, B, C, D, F, G, H},
  {A -> C, A -> D, A -> G, B -> C, B -> D, B -> F, B -> H, C -> F},
  VertexLabels -> "Name"]

```



```

In[16]:= BreadthFirstScan[GR, A, {"PrevisitVertex" -> (Print[#] &)}]
A
C
D
G
B
F
H

```

Figura 2.11: Comando de las búsqueda de profundidad

También podemos el subárbol que resulta del la búsqueda de anchura en Mathematica. Esto se logra con el comando:

```

T=Reap[BreadthFirstScan[g, v, {"FrontierEdge" -> Sow}]] [[2, 1]]
Graph[T, VertexLabels -> "Name"]

```

Y podemos reflejar el subarbol de la búsqueda en el grafo original con el comando:

```

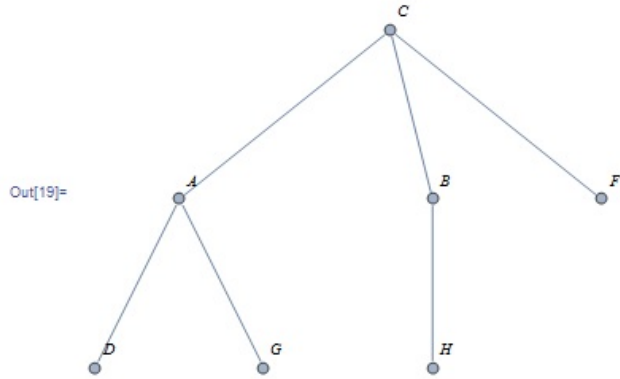
Reap[BreadthFirstScan[g, v, {"FrontierEdge" -> Sow}]] [[2, 1]]
//HighlightGraph[g, #] &

```

En la Figura 2.12 nos muestra la aplicación de estos comandos.

```
T = Reap[BreadthFirstScan[GR, A, {"FrontierEdge" → Sow}]][[2, 1]]
Graph[T, VertexLabels → "Name"]
```

Out[18]= {A → C, A → D, A → G, B → C, C → F, B → H}



```
Reap[BreadthFirstScan[GR, A, {"FrontierEdge" → Sow}]][[2, 1]] // HighlightGraph[GR, #] &
```

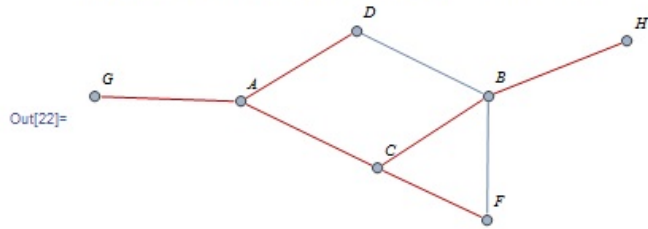


Figura 2.12: Más comandos del algoritmo de profundidad

# Capítulo 3

## Evasividad

En este capítulo hablaremos del concepto de árbol de decisión simple con ello se busca inducir a las funciones evasivas e ilustraremos algunos ejemplos de las mismas, así como funciones no evasivas. Por último representaremos las propiedades de grafos como funciones evasivas para llegar a la conjetura de evasividad con algunos ejemplos. Este capítulo está basado principalmente en las referencias [7] y [8].

### 3.1. Árbol de decisiones y funciones evasivas

**Definición 31.** Un *árbol de decisión* es un árbol que representa la estructura lógica de ciertos algoritmos; esto consta de *nodos* que representan las posibles decisiones que puede tomar el algoritmo dependiendo de la entrada, y las *hojas* representan los posibles resultados que puede tener un algoritmo al pasar por todas las entradas.

Para un árbol de decisión, dada una entrada en particular, uno comienza en la raíz del árbol, este nodo no proviene de una decisión anterior, el cual dependiendo de la entrada se toma la decisión, y desciende en consecuencia en uno de los subárboles de la raíz. Continuando de esta manera, se llega a una hoja que representa el resultado del algoritmo. Por ejemplo, haremos un árbol de decisión que corresponde al algoritmo que da como resultado la nota final a los estudiantes de un curso de cierta institución con duración de 50 días. El promedio de los exámenes y los días asistidos serán las entradas que toma el algoritmo la cual se dividen en cuatro intervalos; de 0.0 a 2.0, de 2.1 a 2.9, de 3.0 a 3.9, y de 4.0 a 5.0. Si el promedio está entre 0.0 a 2.0 ó entre 4.0 a 5.0, el estudiante obtendrá su nota final. Si el promedio está entre 2.6 a 3.5, se evaluará los días asistidos. Si el estudiante asistió mas de 30 días, realizará el recuperatorio. Pero, si esto no ocurre con el estudiante, se dará la nota final. Es claro ver que las hojas son la nota final y el recuperatorio, y los nodos son los intervalos de los promedios de los exámenes y los días asistidos. El diagrama del árbol de decisión de este algoritmo se mostrará en la Figura 3.1.

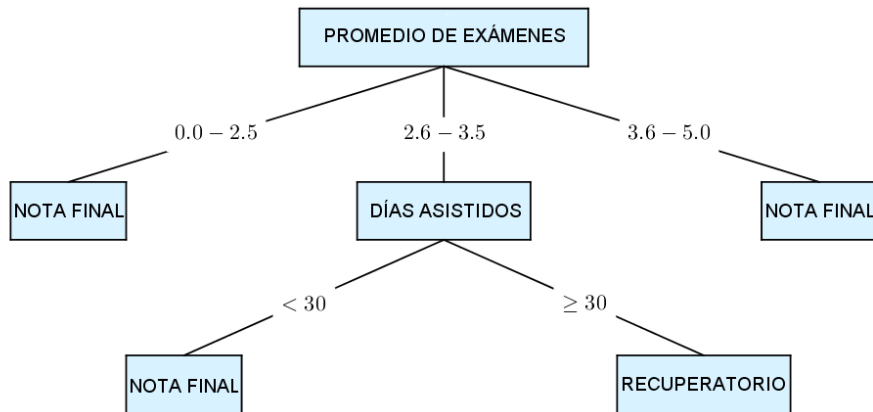


Figura 3.1: Ejemplo de un árbol de decisión

Ahora definiremos un tipo de árbol de decisión que nos vamos a enfocar para el desarrollo de este capítulo.

**Definición 32.** Sea  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , un *árbol de decisión simple* es un árbol binario tal que sus nodos son  $\{1, 2, \dots, n\}$  y sus hojas tienen solo dos valores, 0 y 1.

En un árbol de decisión simple, si el nodo tiene etiqueta  $i$ , realizamos la prueba en  $i$ -ésima entrada; si el resultado es 0 descendimos al subárbol izquierdo, mientras que si el resultado es 1, descendimos al subárbol derecho. El valor de la etiqueta de la hoja resultante es el valor de la entrada de la función. En el siguiente ejemplo mostramos una aplicación de un árbol de decisión simple.

**Ejemplo 4.** Se realiza una clasificación sobre un cultivo de arveja, para este proceso realizamos un árbol de decisión simple. Las entradas del proceso, que a su vez son los nodos, son la longitud de la planta, el color de la flor y el tamaño de la vaina; y las salidas u hojas serían el procesamiento y el no procesamiento de la planta. El árbol de decisión simple de esta clasificación se mostrará en la Figura 3.2.

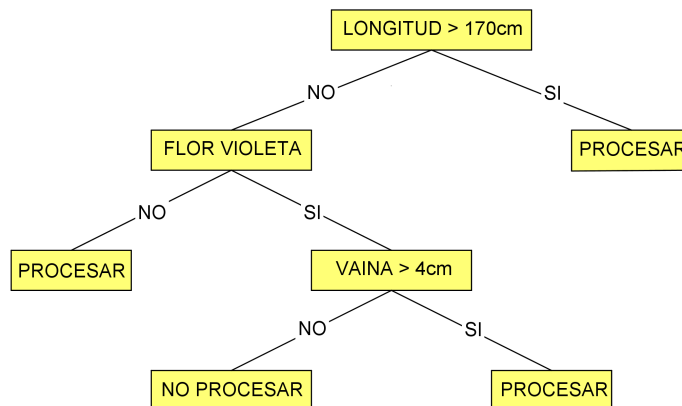


Figura 3.2: Árbol de decisión simple del Ejemplo 4

El Ejemplo 4 podemos verlo como una función  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  si reemplazamos “NO” y “NO PROCESAR” por 0, y “SI” y “PROCESAR” por 1.

Es evidente que cualquier función  $f$  tiene un árbol de decisión simple. Pero, en este trabajo estamos interesados en la siguiente definición.

**Definición 33.** Sea  $f$  una función cualquiera, la *complejidad del árbol de decisión* de  $f$  es la profundidad mínima de un árbol de decisión simple, esto lo denotaremos como  $D(f)$ .

Es claro ver que  $D(f)$  es como máximo el número de variables de  $f$ ; en el peor de los casos la profundidad mínima de un árbol de decisión simple será 0, es decir que tan solo su recorrido será en la raíz. En el siguiente ejemplo veremos como  $D(f)$  alcanza ese número.

**Ejemplo 5.** Sea la función paridad  $f(x_1, \dots, x_n) = x_1 + x_2 + \dots + x_n \pmod 2$ . Para esta función, cualquier árbol de decisión simple  $D(f) = n$  para cada hoja. Ya que, si para algún valor  $x_i$  que no ha sido examinada, llegará a dos valores, 0 y 1. Si es 0, no alterará nada. Pero si es 1, el módulo cambia. De modo que la función calculada no es la paridad. Por lo que es necesario examinar todos los valores de la función.

Con la complejidad de un árbol de decisión para  $f$ , sobresale algunas funciones interesante, las cuales definiremos a continuación.

**Definición 34.** Sea una función  $f$ . Si  $D(f)$  es igual al número de variables de  $f$ , diremos que  $f$  es *evasiva*

Observemos que, la función paridad del Ejemplo 5 es evasiva; veamos un ejemplo no tan trivial de una función evasiva.

**Ejemplo 6.** Sea la matriz

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix}$$

La función  $f$  es 1 si y sólo si en cada fila de la matriz, hay al menos un  $x_{ij}$  que es 1, y 0 en caso contrario, miremos que  $f$  es evasivo. Simulamos un cálculo de algún árbol de decisión. Pero, en lugar de comprobar la matriz, nos preguntamos el valor de cada uno de los  $x_{ij}$ , la cual respondemos 0 siempre y cuando alguna otra variable en la fila permanezca indeterminada y 1 en caso contrario. Siempre mantendremos el valor de la función en indeterminado hasta que todas las variables hallan sido comprobadas. Por lo que la función tiene profundidad mínima  $n^2$ . Así,  $f$  es evasivo.

Ahora, construyamos un ejemplo de una función no evasiva. Sea los jugadores con etiquetas  $1, 2, \dots, n$ , y sea  $x_{ij}$ , con  $1 \leq i < j \leq n$ , que es 1 si el jugador  $i$  le gana al jugador  $j$  y 0 si  $j$  le gana a  $i$ . La función es 1 si hay un jugador que le ganará a todos los demás. Para que  $f$  no sea evasiva, debemos determinar el valor de  $f$  sin jugar todos los encuentros posibles. Para hacer esto, haremos un torneo de todos contra todos. Se

juegan los primeros  $n/2$  juegos. Es decir, 1 con 2, 3 con 4, hasta  $n - 1$  con  $n$ , y tomemos a cualquier jugador  $i$  que haya perdido con algún jugador  $j$ ,  $i$  gana todos sus encuentros restantes, y ahora ponemos a jugar a  $j$  los que les faltan, si  $j$  pierde algún partido, la función es 0, de lo contrario, es 1. Así  $f$  no es evasivo.

### 3.2. La conjetura de evasividad (Aanderaa-Karp-Rosenberg)

Ya teniendo el conocimiento de funciones evasivas, podemos ahora hablar de la conjetura de evasividad, también conocida por los nombres de Stål Aanderaa, Richard M. Karp y Arnold L. Rosenberg.

Podemos representar funciones arbitrarias en grafos, esto se logra con la información que contiene la matriz de adyacencia del grafo, es decir, tomamos un grafo no dirigido  $G$  con  $n$  nodos, por el Capítulo 1 un elemento de la matriz de adyacencia  $x_{ij}$  es 1 si hay una arista entre los vértices  $i$  y  $j$ , y 0 en caso contrario. Nuestro objetivo está direccionado a las funciones que definiremos a continuación.

**Definición 35.** Una *propiedad de grafos* es una función  $f : x_{ij} \rightarrow \{0, 1\}$  con  $1 \leq i < j \leq n$ , si para cualquier  $\sigma \in S_n$  tenemos que  $f(\dots, x_{ij}, \dots) = f(\dots, x_{\sigma(i)\sigma(j)}, \dots)$ . Tengamos en cuenta que  $S_n$  es el grupo de las biyecciones del conjunto  $\{1, \dots, n\}$  sobre sí mismo, los elementos se denominarán permutaciones.

En otros términos, Una propiedad de grafos de  $n$  vértices es una colección  $\mathcal{P}$  de grafos que es cerrada bajo isomorfismo. Un ejemplo de una propiedad de grafos son los grafos conexos de 4 vértices como se ve en la Figura 3.3.

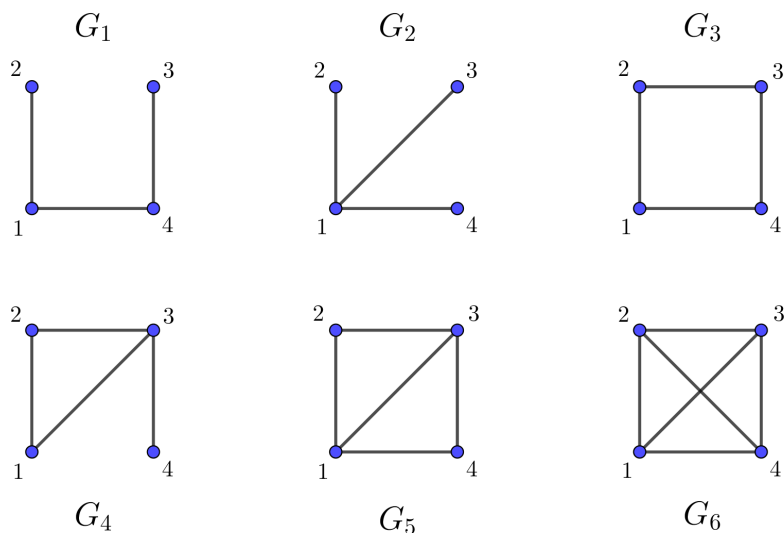


Figura 3.3: Grafos conexos de 4 vértices



Esta colección es cerrada bajo isomorfismos. Además si tomamos el grafo  $H$  de la Figura 3.3 con  $y_{ij}$  un elemento de su matriz de adyacencia podemos encontrar

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 4 \end{pmatrix} \in S_n$$

tal que para cualquier función  $f$ ,  $f(\dots, x_{ij}, \dots) = f(\dots, y_{i,j}, \dots)$   $1 \leq i < j \leq n$ , donde  $x_{ij}$  es un elemento de la matriz de adyacencia del grafo  $G_6$  y  $y_{ij} = x_{\sigma(i)\sigma(j)}$ .

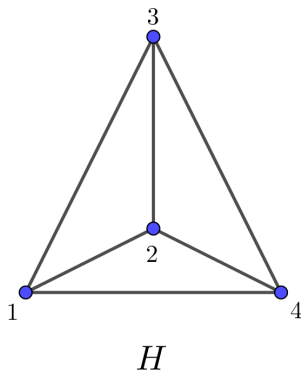


Figura 3.4: El grafo  $H$  es isomorfo al grafo  $G_6$  de la Figura 3.3

Si una propiedad de grafos es cerrada bajo la operación de remover aristas, es decir si un grafo  $G$  está en una propiedad  $\mathcal{P}$ , se le quita cualquier arista  $u$ , entonces  $G - \{u\}$  está en  $\mathcal{P}$ , decimos que esta propiedad es *monótona*. Un ejemplo es la desconexidad, pero la conexidad no lo es, ya que por el Lema 1 si una arista  $u$  es un puente entonces  $G - \{u\}$  tiene dos componentes conexas que es lo mismo que decir que es desconexo. Si una propiedad  $\mathcal{P}$  tiene todos los grafos de  $n$  vértices diremos que es una propiedad *trivial*, de lo contrario es llamada *no trivial*.

Ahora, para hablar de la conjetura de evasividad, consideremos el siguiente juego: Sea  $\mathcal{P}$  una propiedad de grafos en  $n$  vértices. Hay dos personas nombradas como **A** y **B**, ambas conocen la propiedad  $\mathcal{P}$ , la persona **A** conoce un grafo  $G$  que no conoce **B**. El objetivo de la persona **B** es determinar si el grafo  $G$  pertenece o no a  $\mathcal{P}$  siguiendo la siguiente regla: **B** puede hacer preguntas del tipo *¿está la arista  $\{i, j\}$  en el grafo  $G$ ?* a la cual **A** responde SI o NO, dependiendo si la arista está en  $G$  o no, respectivamente. Pero antes, **B** tiene que hacer un listado de preguntas la cual debe tener un orden, esta lista debe tener  $\binom{n}{2}$  preguntas.

Tomamos como ejemplo que la propiedad  $\mathcal{P}$  consiste que los grafos de  $n = 6$  vértices que tiene como máximo 10 aristas. Elegimos  $n = 6$  ya que para  $n \leq 5$ , la propiedad es trivial. Así, **B** tiene una lista de 15 preguntas. Observemos dos casos particulares en la que se desarrolla el juego:

1. Si las primeras 5 preguntas **A** responde NO. **B** le quedan exactamente 10 preguntas por hacer, pero **B** ya puede decir con seguridad que el grafo  $G$  pertenece a  $\mathcal{P}$ , incluso sin saber como es el grafo  $G$ .

2. Si las primeras 10 preguntas **A** responde SI. Inmediatamente después de la pregunta 10 de la lista, **B** puede decir que el grafo  $G$  tiene al menos 10 aristas, si alguna de las 5 preguntas restantes obtiene la respuesta SI, **B** concluye que el grafo  $G$  no está en  $\mathcal{P}$ , la única forma en que **B** pueda concluir que el grafo  $G$  está en  $\mathcal{P}$  es hacer necesariamente las preguntas que le faltan y que todas obtengan respuesta NO, en cuyo caso **B** está obligado a tener que preguntar todas las 15 preguntas; pero, también puede ocurrir que las preguntas 11 a 14 obtenga como respuesta NO, aún es posible que la última tenga respuesta SI, en cuyo caso **B** concluye que el grafo no pertenece a  $\mathcal{P}$ . Pero entonces es posible que **B** esté obligado a tener que hacer todas las 15 preguntas para cumplir su objetivo. Sin importar la lista de preguntas que **B** haga, siempre vamos a encontrar un caso en el que es necesario para **B** hacer el total de 15 preguntas para poder saber si el grafo  $G$  está o no en  $\mathcal{P}$ .

No importa cual sea la lista de preguntas que haga **B** existirá el caso donde **B** esté obligado a realizar las  $\binom{n}{2}$  preguntas para determinar si  $G$  está o no en  $\mathcal{P}$ , lo cual la propiedad  $\mathcal{P}$  es llamada evasiva. En caso contrario,  $\mathcal{P}$  es llamada no evasiva, es decir hay una lista de preguntas para el que sin importar cual sea  $G$ , siempre se puede determinar si  $G$  está o no en  $\mathcal{P}$  antes de efectuar el total de  $\binom{n}{2}$  preguntas.

*La conjetura de evasividad propone que toda propiedad monótona y no trivial de grafos en  $n$  vértices es evasiva.*

Ahora mostraremos algunos ejemplos de propiedades de grafos que cumplen la conjetura.

**Ejemplo 7.** Sea la propiedad de grafos  $\mathcal{P}_k$  que consta de todos los grafos de  $n \geq 6$  vértices que tienen como máximo  $k$  arista. Esta propiedad es monótona ya que a cualquier grafo  $G$  con máximo  $k$  arista, al quitarle cualquier arista tendríamos un grafo de  $k - 1$  aristas.  $\mathcal{P}$  es no-trivial porque no se tiene los grafos desde  $k + 1$  hasta  $\binom{n}{2}$  aristas. Por lo que  $\mathcal{P}$  es evasivo.

**Ejemplo 8.** Las propiedad de grafos desconexos  $\mathcal{P}_{dis}$  es monótono porque al quitarle una arista a un grafo que está en la propiedad, este sigue siendo desconexo; por otro lado es no-trivial. Por lo que  $\mathcal{P}_{dis}$  es evasivo.

**Ejemplo 9.** Según la conjetura, la propiedad de los grafos conexos no es evasiva, ya que la conexidad no es monótona, ya que, por el Lema 1, si tomamos un grafo conexo con una arista como puente, al quitarla del grafo, este deja de ser conexo. Así la propiedad de los grafos conexos no es evasiva

**Ejemplo 10.** Un grafo se le denomina *escorpión* si este posee tres vértices especiales, llamados el *aguijón*, la *cola* y el *cuerpo*. Estos vértice tiene las siguientes condiciones: El aguijón esta conectado solamente con la cola, la cola a su vez, se conecta con el cuerpo, y el cuerpo se conecta con todos los vértices restantes. Un ejemplo de este grafo lo vemos en la Figura 3.5. Sea  $\mathcal{P}_{sco}$  la propiedad de los grafos escorpión, esta propiedad no es evasiva porque a un grafo escorpión, si se le quita la arista que hay entre el aguijón y la cola, este deja de ser escorpión. Por lo que  $\mathcal{P}_{sco}$  no es monótona.

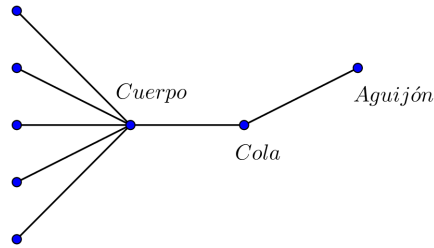


Figura 3.5: Grafo Escorpión

### 3.3. Aplicación en Mathematica

En el Software Mathematica, construimos un programa en el que representa el juego de preguntas en el cual nos ayuda a verificar si el grafo en cuestión pertenece a una propiedad de grafos o no. El programa consiste en dos notebooks. El primer notebook es llamado `Grafos.nb` consiste en que la persona que conoce el grafo lo ingrese (notebook). El notebook lo vemos en la Figura 3.6.

```
V := Input["Ingrese el conjunto de vértices. Ejemplo {1,2,3}"]
A := Input["Ingrese el conjunto de aristas. Ejemplo {1↔2,2↔3,3↔1}"]
G = Graph[V, A, VertexLabels -> "Name" ]
M = AdjacencyMatrix[G]
```

Figura 3.6: Notebook `Grafos.nb`

La persona que hará las preguntas sobre el grafo utilizará el notebook `evasivo.nb`. En este notebook, el conocerá el número de preguntas que puede realizar. Podemos observar el funcionamiento del diagrama de flujo en la Figura 3.7, y vemos con más detalle el notebook en la Figura 3.8.

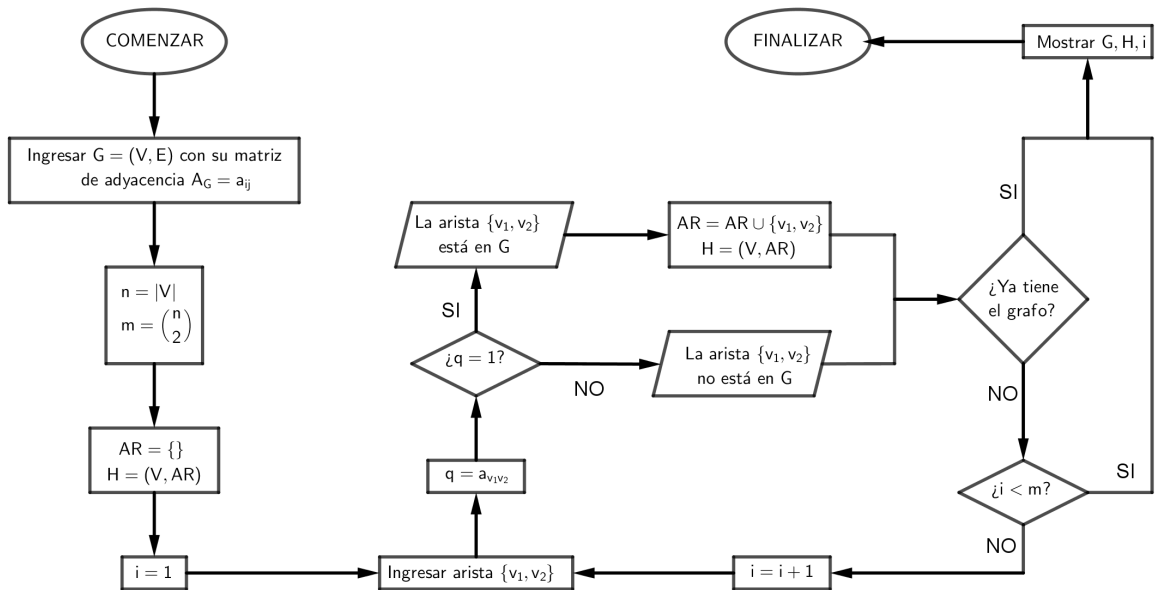


Figura 3.7: Diagrama de flujo del notebook `evasivo.nb`

```

In[11]:= n := VertexCount[G]
         m = EdgeCount[CompleteGraph[n]]

In[32]:= AR := {}
         H := Graph[V, AR, VertexLabels -> "Name"]
         For[i = 1, i ≤ m, i++,
           {v1, v2} = Input["ingrese la pregunta en forma de par de vertices. Ejemplo {n,m}"];
           q = M[[v1, v2]]; If[q == 1, AR = Union[AR, {v1 -> v2}];
           P = Input["La arista existe en el grafo. ¿Ya tiene el grafo?"];
           P = Input["La arista no existe en el grafo. ¿Ya tiene el grafo?"];
           If[P == 1, Abort[]]]
         i
         {G, H}

```

Figura 3.8: Notebook `evasivo.nb`

Realizaremos la prueba de escritorio de este programa. Tomamos el grafo con 5 vértices mostrada en la Figura 3.9.

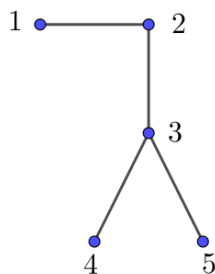


Figura 3.9: Grafo para la prueba de escritorio

Ingresamos el grafo en el notebook `Grafos.nb` por lo que el programa nos muestra como resultado, la imagen del grafo que se muestra en la siguiente Figura 3.10

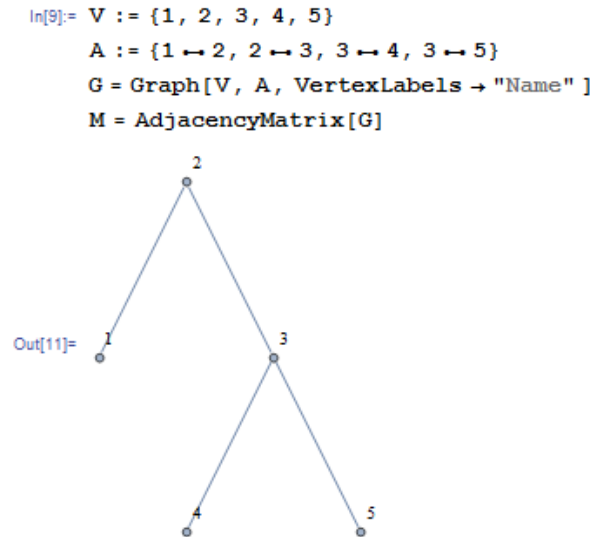


Figura 3.10: Ingreso del grafo en el notebook `Grafos.nb`

Ahora, cerrando el notebook `Grafos.nb` y tomando como propiedad la del escorpión, realizaremos las preguntas en el notebook `evasivo.nb`. El programa nos dirá cuantas preguntas se debe realizar, las cuales son 10 preguntas. Esto se muestra en la Figura 3.11

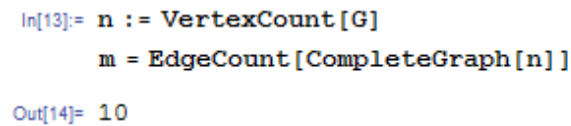


Figura 3.11: Número de preguntas en el notebook `evasivo.nb`

Al correr el resto del notebook, aparecerá un cuadro de diálogo que nos dirá que debemos ingresar una pregunta la existencia de una arista en forma de pareja de vértices. Preguntaremos por la arista que une los vértices 1 y 3, por lo que ingresamos  $\{1, 3\}$ . Así como se muestra en la Figura 3.12

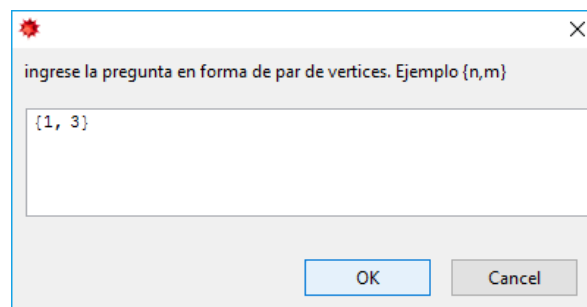


Figura 3.12: Ingreso de la pregunta en el notebook `evasivo.nb`

Al oprimir OK, el programa nos responderá, en este caso, nos dice que la arista que une los vértices 1 y 3 no existe. Luego nos pregunta si ya tenemos el grafo. por lo que respondemos que no, para esto, escribimos 0, como se muestre en la Figura 3.13

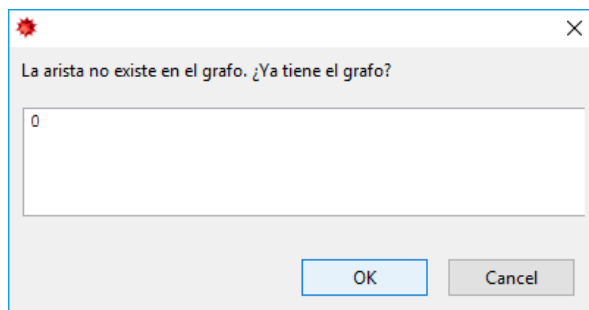


Figura 3.13: Respuesta del programa a la pregunta de la existencia de una arista

Como respondemos NO, el programa nos volverá a pedir que ingresemos otra pregunta con el cuadro de diálogo que se mostró en la Figura 3.12, y este nos volverá a responder si la arista está en el grafo o no, y nos volverá a preguntar si ya tenemos el grafo. Este ciclo seguirá hasta que se realice el número total de preguntas, esto pasa por no se logra saber si el grafo está en la propiedad. Si en la  $i$ -ésima pregunta respondida por el programa se sabe si el grafo está o no en la propiedad, responderemos SI como 1 en el cuadro de diálogo de la Figura 3.13 y el programa parará de hacer las preguntas.

Para la prueba de escritorio, realizamos todas las preguntas. Por último el programa nos mostrará cuantas preguntas realizamos y nos mostrará dos grafos  $G$  y  $H$ , el primero será el grafo que se estuvo preguntando, y el segundo será el grafo que construimos a partir de las preguntas. Si hacemos todas las preguntas, los grafos deberán ser iguales; pero si no se realiza todas las preguntas, el segundo grafo será un subgrafo extendido del primer grafo. Los resultados del programa los mostramos en la Figura 3.14

```

In[5]:= n := VertexCount[G]
        m = EdgeCount[CompleteGraph[n]]

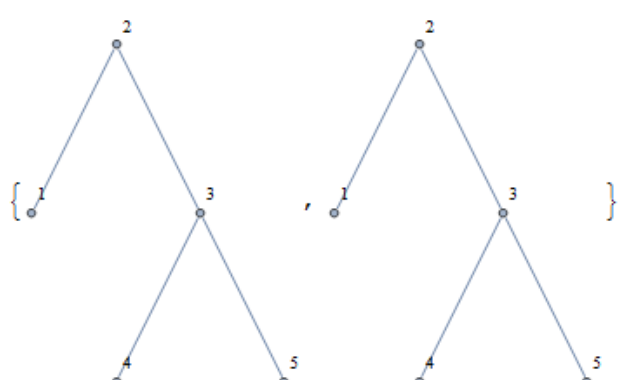
Out[6]= 10

In[35]:= AR := {}
H := Graph[V, AR, VertexLabels -> "Name"]
For[i = 1, i ≤ m, i++,
  {v1, v2} =
    Input["ingrese la pregunta en forma de par de vertices. Ejemplo {n,m}"];
  q = M[{v1, v2}]; If[q == 1, AR = Union[AR, {v1 ↔ v2}];
  P = Input["La arista existe en el grafo. ¿Ya tiene el grafo?"];
  P = Input["La arista no existe en el grafo. ¿Ya tiene el grafo?"];
  If[P == 1, Abort[]]]

Out[37]= $Aborted

In[38]:= i
        {G, H}

Out[38]= 10

Out[39]= {

}

```

Figura 3.14: Resultados del notebook `evasivo.nb`

Con ello concluye la aplicación realizada por nosotros de algunos casos particulares en la cual se pudiese verificar si una propiedad es evasiva.

# Conclusiones


- Durante este trabajo de grado realizamos un recorrido en la teoría de grafos y sus aplicaciones lo cual nos ayudó a dar forma al concepto de la conjetura de evasividad en grafos, a través de la identificación de las principales definiciones y formas de clasificar grafos que han sido muy relevantes para el desarrollo de la teoría.
- La comprensión de los principales algoritmos de recorrido sobre grafos resultaron de gran importancia dado que permitieron visionar el trabajo computacional sobre la teoría de grafos, además con ayuda del software Mathematica se pudo validar algunos ejemplos.
- Se encontró en la bibliografía la definición matemática de la conjetura de evasividad basada en las definiciones de árbol de decisión, algoritmo de profundidad y funciones evasivas; la comprensión de la misma resultó ser un proceso práctico por la revisión de los conceptos y los algoritmos que se realizaron previamente.
- Se diseñó un algoritmo que ayuda a verificar si una propiedad de grafos es evasiva, el cual está basada principalmente por la matriz de adyacencia que a su vez hace parte de las bases de los algoritmos de profundidad y anchura; de dicho algoritmo se formuló su diagrama de flujo para posibilitar la implementación en software, para este trabajo de grado su implementación fue en el software Mathematica.
- A manera de recomendación sugerimos que se continúe realizando estudios acerca de la conjetura ya que aún se considera un problema abierto con aplicabilidad en las ramas de las ciencias de la computación y naturalmente en las matemáticas.



# Bibliografía

- [1] Araúz Lombardía, Cristina.. *Problemas y conjeturas de la teoría de grafos*. Barcelona (España): Universidad de Barcelona, Trabajo académicamente dirigido para la Licenciatura de Matemáticas, 2004,2009.
- [2] Cardoso Moreira, Domingos.. *Teoria dos Grafos e Aplicações*. Sociedade Brasileira de Matemática, 2004.
- [3] Fernández Gallardo, Pablo y Fernández Pérez, José Luis. *Notas de Matemática Discreta*. Madrid (España): Universidad Autónoma de Madrid, 2004.
- [4] Gross, Jonathan L., Yellen, Jay and Zhang, Ping. *Handbook of Graph Theory*, Second edition. Chapman & Hall/CRC, 2013.
- [5] Lipschutz, Seymour., Lipson, Marc, *Matemáticas discretas*. Charlottesville (Virginia): University of Virginia, Filadelfia (Pensilvania): Temple University, 1976.
- [6] Liu, C. L., *Introduction to Combinatorial Mathematics*. New York: McGraw-Hill, 1968.
- [7] Lovász László and Young Neal E. *Lecture Notes on Evasiveness of Graph Properties*. Princeton University, 1990.
- [8] Lutz, Frank H.. *Some results related to the evasiveness conjectures*, Journal of Combinatorial Theory Series B, v.81 n.1, p.110–124, Jan, 2001.
- [9] Wolfram, Stephen.. *An elementary to the Wolfram language*, Manitoba (Canada): Acid-free paper, 2016.



 Universidad del Tolima	<b>PROCEDIMIENTO DE FORMACIÓN DE USUARIOS</b>  <b>AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL</b>	Página 2 de 3
		Código: GB-P04-F03
		Versión: 03
		Fecha Aprobación: 15 de Febrero de 2017

Producto de la actividad académica/científica/cultural en la Universidad del Tolima, para que con fines académicos e investigativos, muestre al mundo la producción intelectual de la Universidad del Tolima. Con todo, en mi condición de autor me reservo los derechos morales de la obra antes citada con arreglo al artículo 30 de la Ley 23 de 1982. En concordancia suscribo este documento en el momento mismo que hago entrega del trabajo final a la Biblioteca Rafael Parga Cortes de la Universidad del Tolima.

De conformidad con lo establecido en la Ley 23 de 1982 en los artículos 30 “...**Derechos Morales. El autor tendrá sobre su obra un derecho perpetuo, inalienable e irrenunciable**” y 37 “...**Es lícita la reproducción por cualquier medio, de una obra literaria o científica, ordenada u obtenida por el interesado en un solo ejemplar para su uso privado y sin fines de lucro**”. El artículo 11 de la Decisión Andina 351 de 1993, “**los derechos morales sobre el trabajo son propiedad de los autores**” y en su artículo 61 de la Constitución Política de Colombia.

- Identificación del documento:

Título completo: **ELEMENTOS DE LA TEORÍA DE GRAFOS Y LA CONJETURA DE EVASIVIDAD**

- Trabajo de grado presentado para optar al título de:

**PROFESIONAL EN MATEMATICAS CON ENFASIS EN ESTADISTICA**

- Proyecto de Investigación correspondiente al Programa (No diligenciar si es opción de grado “Trabajo de Grado”):

\_\_\_\_\_

- Informe Técnico correspondiente al Programa (No diligenciar si es opción de grado “Trabajo de Grado”):

\_\_\_\_\_

- Artículo publicado en revista:

\_\_\_\_\_

- Capítulo publicado en libro:

\_\_\_\_\_

- Conferencia a la que se presentó:

\_\_\_\_\_

