



UNIVERSIDAD DE ORIENTE
NÚCLEO DE SUCRE
ESCUELA DE CIENCIAS
DEPARTAMENTO DE INFORMÁTICA

APLICACIÓN PARA DETERMINAR LA HAMILTONICIDAD DE GRAFOS
BIPARTITOS BALANCEADOS
(Modalidad: Tesis de Grado)

MARÍA DE LOS ÁNGELES ALZOLAR MARCANO

TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA
OPTAR AL TÍTULO DE LICENCIADO EN INFORMÁTICA

CUMANÁ, MARZO 2017

APLICACIÓN PARA DETERMINAR LA HAMILTONICIDAD DE GRAFOS
BIPARTITOS BALANCEADOS

APROBADO POR:

Profa. Carmen Romero
Asesor Académico

Prof. Daniel Brito
Co-Asesor Académico

Jurado

Jurado

ÍNDICE

AGRADECIMIENTOS	i
DEDICATORIA	ii
LISTA DE TABLAS	iii
LISTA DE FIGURAS.....	iv
RESUMEN	v
INTRODUCCIÓN	1
CAPÍTULO I. PRESENTACIÓN	4
1.1 Planteamiento del problema.....	4
1.2 Alcance y limitaciones	8
1.2.1 Alcance	8
1.2.2 Limitaciones	8
CAPÍTULO II. MARCO DE REFERENCIA	9
2.1 Marco teórico	9
2.1.1 Antecedentes de la investigación.....	9
2.1.2 Bases teóricas.....	11
2.2 Marco metodológico	15
2.2.1 Metodología de la investigación	15
2.2.2 Metodología del área aplicada	15
CAPÍTULO III: DESARROLLO	21
Iteración 1	21
Identificación, observación y planteamiento del problema	21
Construcción del modelo	29
Generación de una solución.....	30
Prueba y evaluación de una solución	30
Iteración 2	31
Identificación, observación y planteamiento del problema	31
Generación de una solución.....	32
Prueba y evaluación de una solución.....	35
Implante	36
Iteración 3	36

Generación de una solución.....	36
Prueba y evaluación de una solución.....	40
Iteración 4	41
Generación de una solución.....	41
Implante	41
Evaluación	43
CONCLUSIONES	44
RECOMENDACIONES.....	45
BIBLIOGRAFÍA	46
APÉNDICES	47
ANEXOS	66
HOJA DE METADATOS	82

AGRADECIMIENTOS

Agradezco a:

Mis padres, Carmen Marcano, Santos Alzolar, por todo el apoyo que recibí de su parte, desde el inicio de mi carrera y hasta estos momentos, gracias.

Mis asesores, Carmen Romero y Daniel Brito, quienes me ofrecieron la oportunidad de trabajar en esta hermosa investigación y siempre estuvieron dispuestos a brindarme su ayuda, conocimiento, y asesoramiento para llegar a feliz término con este trabajo de investigación, gracias.

A mi mejor amigo y compañero, Héctor Alonso por sus consejos y por el apoyo que me brindo durante mi carrera y durante esta investigación, gracias.

A todos los profesores de la Licenciatura en Informática por haberme formado como un profesional, simplemente gracias.

A mis compañeros, con quienes compartí durante el transcurso de mi carrera y me dejaron enseñanzas.

DEDICATORIA

Este trabajo se lo dedico a:

Dios, por haberme dado salud y toda la fuerza que le pedí para continuar y cumplir mi meta.

Mis padres, Carmen Leonides, Santos Emilio por todas sus enseñanzas y el ejemplo que me han dado para ser una persona de bien.

A mis hermanos Luis Emilio, Francisco Javier, Carminia de los Ángeles.

A mis sobrinos, que me llenan de alegría y me inspiran a ser un buen ejemplo para ellos.

LISTA DE TABLAS

Tabla 1. Lista de actividades para el diagrama PERT/CPM.....	23
Tabla 2. Lista de actividades para el diagrama PERT/CPM con duración en semanas y precedencia.	24
Tabla 3. Requisitos del sistema.....	26
Tabla 4. Requisitos del sistema y su clasificación.....	27

LISTA DE FIGURAS

Figura 1. Conjunto independiente balanceado de G con $R = 6$	5
Figura 2. Conjunto independiente balanceado de G con $R = 4$	6
Figura 3. Conjunto independiente balanceado de G con $R = 4$ con $q < p$	7
Figura 4. Conjunto independiente balanceado de G con $R = 4$ con $p < q$	7
Figura 5. El grafo minimal de $H14$	10
Figura 6. Proceso de solución de problemas de CA/IO	20
Figura 7. Árbol de decisión.....	22
Figura 8. Diagrama PERT/CPM.....	25
Figura 9. Diagrama de casos de uso del análisis de requisitos	28
Figura 10. Diagrama de Casos de uso Gestionar Grafo.....	29
Figura 11. Diagrama de clases preliminar del modelo matemático.....	30
Figura 12. Prueba de grafo generado mediante la consola de comando de Python.....	31
Figura 13. Diseño de interfaz preliminar para la aplicación.....	31
Figura 14. Error de sobre carga de memoria durante las permutaciones para grafos de mínimo grado 5.....	35
Figura 15. Vista de interfaz para el módulo de los grafos	40
Figura 16. Intérprete de grafos.....	40
Figura 17. Resultado del intérprete al ingresar aristas y vértices erróneos.....	41
Figura 18. Resultado del intérprete al ingresar vértices erróneos.....	41
Figura 19. Resultado de la aplicación al graficar un grafo	42
Figura 20. Resultado de la aplicación con un grafo cualquiera ingesado.....	42
Figura 21. Resultados de la aplicación al graficar un grafo.....	43

RESUMEN

Se desarrolló una aplicación para determinar la hamiltonicidad de grafos bipartitos balanceados en función de su mínimo grado y número de vecinos, ofreciendo funcionalidades para manipular grafos de características específicas, entre ellas; grafos simples y de orden $2n$. Para esta investigación se tomó como referencia la metodología CA/IO (Ciencia de la Administración e Investigación de Operaciones) la cual consta de 6 fases: identificación, observación y planteamiento del problema; donde se obtuvo los requisitos de la aplicación estudiando el modelo matemático planteado. Construcción del modelo, en donde se plantearon las soluciones del problema en cuestión, y se planteó un árbol de decisión con los pasos a seguir. Generación de una solución, en el cual se construyó los algoritmos de cada proceso de la aplicación guiándose por el árbol de decisión planteado. Prueba y evaluación de una solución, en el que se evaluaron los procesos y se dio fe de su veracidad. Implante y evaluación. Se implementó PERT/CPM para la planificación de la misma. La aplicación obtenida ofrece a investigadores y a estudiantes del área de estudio una herramienta para facilitar el estudio de los grafos bipartitos y mejorar el desempeño de los mismos al agilizar el proceso.

Palabras claves: aplicación, hamiltonicidad, teoría de grafos, grafos bipartitos balanceados, conjunto independiente de vértices.

INTRODUCCIÓN

La teoría de grafos juega un papel muy importante en las diferentes áreas de estudio, ya que todas sus aplicaciones se utilizan en la vida cotidiana, como en líneas telefónicas, líneas de televisión por cable, redes de computadoras, entre otros. Según Giudici & Bris (1997, p.3):

La teoría de grafos es aplicable a todos los tipos de áreas de estudio, como lo son: física, química, comunicación, informática, ingeniería civil y eléctrica, arquitectura, investigación de operaciones, genética, psicología, sociología, economía, antropología y lingüística.

El comienzo de la teoría de grafos se atribuye a Euler en el año 1736, quien resolvió el problema de los puentes de Königsber, basándose en el concepto de grafos. Luego, en 1847, Kirchoff empleó la teoría de grafos para el análisis de las redes eléctricas. Es considerado el primer caso en que se aplica la teoría de grafos en un problema de ingeniería. No obstante, el problema de los cuatro colores planteado en 1852 por Guthrie, fue lo que arrastró a más jóvenes al estudio de teoría de grafos. (Giudici & Bris, 1997).

En el ámbito de la informática la teoría de grafos puede servir para el modelado de sistemas, la simulación, la estructuración de datos y el análisis y diseño de algoritmos. Gracias a la teoría de grafos se han desarrollado una gran variedad de algoritmos y métodos de resolución eficaces que permiten tomar una mejor decisión, entre ellos conocemos algoritmos como Dijkstra, Kruscal, y otros.

Caicedo, Wagner & Méndez (2010, p.1), expresan que (...) “un grafo G consta de un conjunto de vértices o nodos V y un conjunto de arcos A , cada uno de los cuales une un vértice con otro. $G = (V, A) = (N, A)$ ”.

Caicedo *et al.* (2010, p.1) enuncia lo siguiente:

Los arcos que unen los nodos también se llaman aristas del grafo y se representan por medio de un par de elementos (v_i, v_j) , donde los

elementos son los nodos que une al arco. (...) El número de nodos de un grafo se llama orden del grafo y se denota como: $\text{Ord } g = N_G = N$.

Existe una variedad de conceptos que se manejan en la teoría de grafos, para ésta investigación en cuestión es importante conocer los siguientes: camino, ciclo, camino hamiltoniano, ciclo hamiltoniano, grafos bipartitos, grafos bipartitos balanceados, conjunto independiente, mínimo grado.

Schmidt & Mendieta (2002, p.30) señalan:

Un camino en un grafo G es una secuencia de nodos adyacentes, en la que cada uno aparece una sola vez, junto con las aristas definidas por los pares de nodos adyacentes en la secuencia. (...) Un ciclo como un camino más una arista que se encuentra entre los nodos inicial y final.

En ocasiones se pueden apreciar grafos bipartitos. En estos grafos el conjunto de vértices V se puede dividir en dos conjuntos disjuntos V_1, V_2 , tales que, $V_1 \cup V_2 = V$ y $V_1 \cap V_2 = \emptyset$, de tal manera que toda arista $e \in E$ conecta un vértice de V_1 con un vértice de V_2 . (Giudici & Bris, 1997, p.11).

Chacón (2005, p.26), también define un camino hamiltoniano de G como (...) “Un camino simple que contiene cada vértice de G ; análogamente, un ciclo hamiltoniano de G es un ciclo que contiene todos los vértices de G .”

Si el grafo posee un ciclo hamiltoniano entonces se dice que es un grafo hamiltoniano. Los grafos bipartitos al igual que los grafos en general, son de gran utilidad ya que cuentan con muchas propiedades que han ayudado al estudio de problemas basados en la programación lineal, topología, computación, entre otros. De igual forma que los grafos hamiltonianos.

Es relevante saber que dos vértices de un grafo G son adyacentes si existe una arista que los une. Un conjunto de vértices independiente de un grafo G , es un conjunto S de vértices de G tal que no existen dos vértices adyacentes contenidos en S . El mínimo grado de un vértice de un grafo G es el menor número de lados adyacentes a él. En

grafos bipartitos balanceados un conjunto independiente balanceado es un conjunto de vértices independientes con igual número de vértices de cada partición.

CAPÍTULO I. PRESENTACIÓN

1.1 Planteamiento del problema

El problema de hamiltonicidad es un problema Np-Completo, es decir; un problema que no ha sido resuelto en tiempo polinomial. Por lo tanto, muchos investigadores en el área se han dedicado al estudio de problemas de este tipo, aunque no exista alguna implementación óptima para determinar la hamiltonicidad de un grafo. En este sentido, un grupo de investigadores representado por el Licenciado en Matemática Daniel Brito; en la Universidad de Oriente Núcleo de Sucre, han creado y validado una teoría matemática que permite determinar la hamiltonicidad de grafos bipartitos balanceados en función de su mínimo grado, y número de vecinos. Esto para encontrar una forma más aligera de determinar cuándo un grafo bipartito es hamiltoniano, esta teoría fue planteada en el año 1998 para grafos de mínimo grado 3 y es la siguiente:

Sea G un grafo bipartito balanceado de orden $2n$, es decir, un grafo con una bipartición en dos conjuntos de vértices independientes cada uno de ellos con n vértices, y mínimo grado, es decir, el menor número de lados incidentes a un vértice, al menos tres. Si para cada conjunto independiente balanceado de cuatro vértices, es decir, un conjunto independiente conteniendo dos vértices de cada lado de la bipartición, su número de vecinos es al menos $n + 1$, entonces G es hamiltoniano.

Nos queda el siguiente lema que llamaremos Lema 1:

Sea G un grafo bipartito balanceado de orden $2n$ y mínimo grado al menos tres. Si para cada conjunto independiente balanceado de cardinalidad cuatro su número de vecinos es al menos $n + 1$, entonces G es hamiltoniano.

Esta teoría en cuestión se encuentra validada según el grado del grafo y número de vecinos, a continuación se mostrará el primer lema de la teoría anteriormente enunciada para grafos de grado mínimo tres (3):

Sea G un grafo bipartito balanceado no hamiltoniano de orden $2n$ y mínimo grado $\delta(G) \geq 3$. Si P es un camino hamiltoniano de G , entonces existe un conjunto independiente balanceado de P tal que $|S| = 4$.

Prueba: sea $\{A, B\}$ una bipartición balanceada de $V(G)$ y P un camino hamiltoniano de G . Sin pérdida de generalidad sea $P = a_1 b_1 a_2 b_2, \dots, a_n b_n$ tal camino de G . Como G tiene mínimo grado 3, entonces a_1 y b_n tienen al menos 3 vecinos en P . Si hay dos vértices $b_i \in N(a_1)$ y $a_j \in N(b_n)$ tal que $i > j$, entonces P es llamado un camino cruzado. En otro caso P es llamado un camino no cruzado.

Caso 1: P es un camino no cruzado

Como P es un camino no cruzado $a_1 b_n \notin E(G)$. Sin pérdida de generalidad, sean $R_1 = \{b_i \in V(P) : b_i \in N(a_1), 2 \leq i \leq q\}$ y $R_2 = \{a_i \in V(P) : a_i \in N(b_n), q < p \leq j \leq n-1\}$, los conjuntos de vecinos de a_1 y b_n respectivamente. Por lo tanto $R = \{b_i^- \in V(P) : b_i \in R_1\} \cup \{a_i^+ \in V(P) : a_i \in R_2\} \cup \{a_1, b_n\}$ es un conjunto independiente balanceado de G con $|R| \geq 6$, en consecuencia existe un conjunto independiente balanceado $S \subseteq R$ tal que $|S| = 4$. Ver Figura 1.

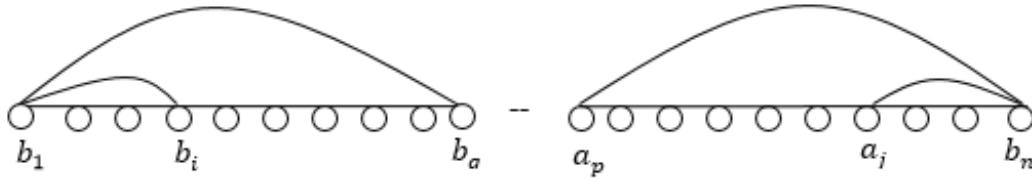


Figura 1. Conjunto independiente balanceado de G con $|R| = 6$

Caso 2: P es un camino cruzado

Como P es un camino cruzado, hay vértices $b_i \in N(a_1)$ y $a_j \in N(b_n)$ tal que $i > j$. Sin pérdida de generalidad, sean $i = \max\{1 : b_1 \in N(a_1)\}$ y $j = \min\{1 : a_1 \in N(b_n)\}$, entonces $b_i \in N(a_1)$ y $a_j \in N(b_n)$.

Sean $I_1 = P[a_2, b_{j-1}]$, $I_2 = P[a_j, b_i]$ y $I_3 = P[a_{i+1}, b_{n-1}]$

Como G tiene mínimo grado $\delta G \geq 3$, entonces a_1 y b_n tienen ambos al menos dos vecinos más en P , permítase con estos formar los conjuntos no vacío $R_1 = \{b_1 \in V(P) : b_1 \in N^-(a_1)\}$ y $R_2 = \{a_1 \in V(P) : a_1 \in N^-(b_n)\}$.

Subcaso 2.1: $R_1 \cap I_1 \cup I_2 \neq \emptyset$ y $R_2 \cap I_2 = \emptyset$

Entonces $R = \{b_1^- \in V(P) : b_1 \in R_1 \cap I_1 \cup I_2\} \cup \{a_1^+ \in V(P) : a_1 \in R_2 \cap I_3\} \cup \{a_1, b_n\}$ es un conjunto independiente balanceado de G con $|R| \geq 4$, y consecuentemente existe un conjunto independiente balanceado $S \subseteq R$ tal que $|S| = 4$. Ver Figura 2.

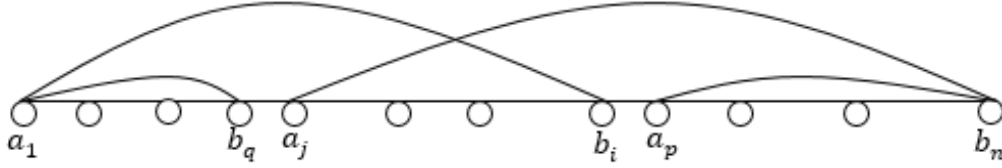


Figura 2. Conjunto independiente balanceado de G con $|R| = 4$.

Subcaso 2.2: $R_1 \cap I_1 = \emptyset$ y $R_2 \cap I_3 = \emptyset$

Como G tiene mínimo grado $\delta G \geq 3$, entonces $|R_1 \cap I_2| \geq 1$ y $|R_2 \cap I_2| \geq 1$. Sin pérdida de generalidad, supóngase, $b_q \in R_1 \cap I_2$ y $a_p \in R_2 \cap I_2$. Se divide en dos subcasos.

Subcaso 2.2.1: $q < p$

Entonces $R = \{b_1^- \in V(P) : b_1 \in R_1 \cap I_2, 1 \leq q\} \cup \{a_1^+ \in V(P) : a_1 \in R_2 \cap I_2, p \leq 1\} \cup \{a_1, b_n\}$ es un conjunto independiente balanceado de G con $|R| \geq 4$, y consecuentemente existe un conjunto independiente balanceado $S \subseteq R$ tal que $|S| = 4$. Ver Figura 3.

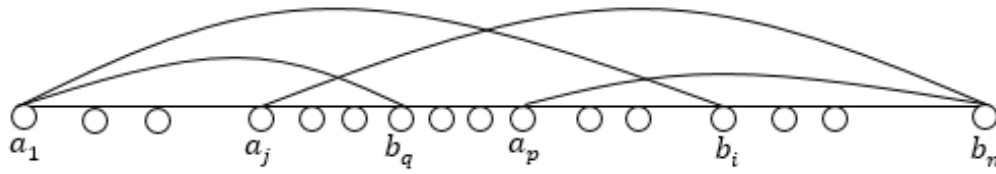


Figura 3. Conjunto independiente balanceado de G con $R = 4$ con $q < p$.

Subcaso 2.2.2: $p < q$

Entonces $R = \{b_1^+ \in V : b_1 \in R_1 \cap I_2, p < 1 \leq q\} \cup \{a_1^+ \in V : a_1 \in R_2 \cap I_2, 1 \leq p\} \cup \{a_j^+, b_i^+\}$ es un conjunto independiente balanceado de G con $R \geq 4$, y consecuentemente existe un conjunto independiente balanceado $S \subseteq R$ tal que $|S| = 4$. Ver Figura 4.

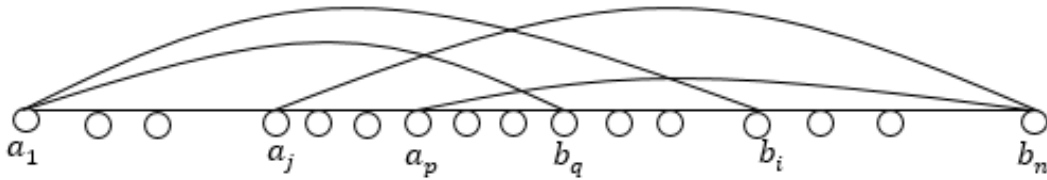


Figura 4. Conjunto independiente balanceado de G con $R = 4$ con $p < q$

En todos los casos existe un conjunto independiente balanceado S de cardinalidad 4, y el lema es probado.

La cantidad de grafos que contengan las características necesarias para la validación de la teoría planteada puede ser significativa y; además de ello, los mismos pueden tener un tamaño considerable, esto ocasiona mayor complejidad y más tiempo en el momento de llevar a cabo las validaciones correspondientes.

Debido a lo anteriormente planteado, y en pro de seguir con los avances tecnológicos en el área de grafos, surgió la intención de realizar ésta investigación, la cual consiste en el desarrollo de una aplicación que determine la hamiltonicidad de un grafo bipartito balanceado en función de su mínimo grado y número de vecinos, basándose en el lema 1 anteriormente expresado.

1.2 Alcance y limitaciones

1.2.1 Alcance

Se cumplió con cada uno de los requerimientos que se derivaron del estudio del modelo matemático apoyado en las fases que proporciona la metodología de investigación CA/IO.

La aplicación puede ser utilizada para validar el modelo matemático planteado y está en la capacidad de realizar lo siguiente:

Calcular el mínimo grado de un grafo.

Generar gráfico de los grafos ingresados.

Determinar la bipartición balanceada de un grafo dado.

Interpretar la entrada de los grafos en forma matemática.

Interpretar la entrada de los grafos en forma gráfica.

Determinar la hamiltonicidad de grafos bipartitos balanceados en función de su mínimo grado y número de vecinos.

1.2.2 Limitaciones

La capacidad de cálculo en término de cantidad de vértices, es decir; grado del grafo, depende de los recursos que disponga el computador; como la cantidad de memoria principal y procesamiento.

Las pruebas unitarias realizadas con la aplicación, alcanzó a validar el modelo con grafos de grado mínimo 5. Los algoritmos de la aplicación tienen una estructura secuencial.

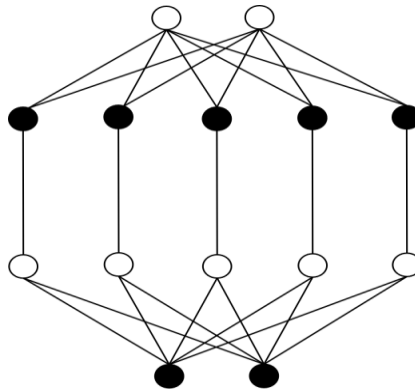
CAPÍTULO II. MARCO DE REFERENCIA

2.1 Marco teórico

2.1.1 Antecedentes de la investigación

La investigación de ciertos problemas extremales envolviendo una condición de unión de vecindades para conjuntos independientes balanceados de cardinalidad cuatro en grafos bipartitos balanceados fue iniciada por Amar, Brandt, Brito y Ordaz en 1998.

Amar, Brandt, Brito & Ordaz (1998), poseen la siguiente conjetura: “Sea G un grafo bipartito balanceado de orden $2n$ y mínimo grado al menos tres. Si para cada conjunto independiente balanceado de cardinalidad cuatro su número de vecinos es al menos $n + 1$, entonces G es hamiltoniano ó $G \in H_{14}$.” Donde H_{14} denota la clase de grafos obtenido del grafo dibujado en la figura 5, donde alguno (o todos) de los cuatro lados posibles, uniendo los vértices de la parte superior con los vértices de la parte inferior pueden estar presentes.



Amar *et al.* (1998), probaron: “Sea G un grafo bipartito balanceado de orden $2n$ y mínimo grado al menos tres. Si para cada conjunto independiente balanceado de cardinalidad cuatro su número de vecinos es al menos $n + 3$, entonces G es hamiltoniano”.

Figura 5. El grafo minimal de H_{14} .

Tratando de probar la conjetura los profesores Brito & Lárez (2007), demostraron: “Sea G un grafo bipartito balanceado de orden $2n$ y mínimo grado al menos cuatro. Si para cada conjunto independiente balanceado de cardinalidad cuatro su número de vecinos es al menos $n + 1$, entonces G es hamiltoniano”.

Los profesores Alcalá, Brito & Marín (2013), demostraron: “Sea G un grafo bipartito balanceado de orden $2n$ y mínimo grado al menos cinco. Si para cada conjunto independiente balanceado de cardinalidad seis su número de vecinos es al menos $n + 2$, entonces G es hamiltoniano”.

Se han realizado muchas investigaciones referentes al área de teoría de grafos, se tiene el trabajo de García (2012) intitulado: “Aplicación para determinar si una sucesión de números es bipartita gráfica”, como requisito para optar por el título de Licenciado en Informática.

Esta aplicación se desarrolló para la automatización y representación gráfica de grafos bipartitos balanceados, y busca ayudar a resolver instancias de problemas que se creían tediosos de graficar mediante las representaciones gráficas generadas por la aplicación.

2.1.2 Bases teóricas

Coss (2003), define Simulación como "una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos períodos de tiempo".

Este trabajo de grado se enmarca dentro del área de los sistemas de simulación, fundamentado en el uso del computador y otras tecnologías informáticas para la automatización.

A continuación, se presentan una serie de conceptos relacionados con dicha área y conceptos básicos los cuales fueron necesarios para el desarrollo de esta aplicación.

Un algoritmo es un conjunto de procedimientos o reglas que, cuando se siguen de forma ordenada, proporcionan la mejor solución para un modelo determinado.

Modelo matemático es la representación simbólica y abstracta de un problema.

Software Libre es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido el mismo, puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas.

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto lo conceptuales (procesos del negocio y funciones del sistema) como lo concreto (esquemas de bases de datos y componentes de software reutilizables) (Booch y cols., 2004).

El diagrama de casos de uso muestra las relaciones existentes entre actores y casos de uso dentro de un sistema. Un caso de uso describe una interacción con los actores como secuencia de mensajes entre el sistema y uno o más actores, este es representado gráficamente por una elipse. El término actor incluye a los seres humanos, así como a otros sistemas informáticos y procesos.

El diagrama de clases es una presentación gráfica de la vista estática de un sistema que muestra una colección de elementos declarativos (estáticos) del modelo, como clases, que son una descripción generalizada o categorización de un grupo de cosas que tienen atributos y acciones similares, encapsulan los datos y las abstracciones de procedimientos requeridos para describir el contenido y el comportamiento de alguna entidad del mundo real, tipos y sus contenidos y relaciones (Booch y cols., 2004).

Las aplicaciones Web, también llamadas soluciones Intranet, son sistemas capaces de encargarse del acceso a la información de una organización y sus procesos, así como de la distribución de esta información y de las necesidades de comunicación, para crear, editar, manipular o cambiar los datos de la Intranet (Powell, 2001).

El lenguaje de programación utilizado para la creación de la aplicación para determinar la hamiltonicidad de grafos bipartitos balanceados fue Python el cual fue creado a finales de los ochenta por Guido van Rossum, administrado por la Python Software Foundation. Posee una licencia de código abierto denominada Python Software Foundation License. Se trata de un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible, usa tipado dinámico y es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

NetworkX es un paquete de software del lenguaje Python para la creación, manipulación, y el estudio de la estructura, la dinámica y la función de las redes complejas. Nació en mayo de 2002, la versión original fue diseñada y escrita por Aric Hagberg, Dan Schult, y Pieter Swart en 2002 y 2003, es software libre, se puede redistribuir y/o modificar bajo los términos de la licencia BSD.

Django es un framework Web de Python de alto nivel que fomenta el rápido desarrollo y diseño limpio y pragmático. Es gratuito y de código abierto.

El comportamiento estático se realiza mediante el lenguaje de marcado de hipertexto HTML, el cual es una implementación del Standard Generalized Markup Language (SGML), estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones.

HTML es un metalenguaje para definir lenguajes de diseño descriptivos; proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones. La descripción se basa en especificar en el texto la estructura lógica del contenido como títulos, párrafos, enumeraciones, así como los diferentes efectos que se quieren dar (cursiva, negrita, subrayado) y dejar que luego la presentación final de dicho hipertexto se realice a través de programas especializados conocidos como navegadores Web (W3C, 2010).

CSS, hojas de estilo en cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos (W3C, 2015).

Un script es el código del programa que no necesita pre-procesamiento, como la compilación antes de ejecutarse. En el contexto de un navegador Web, scripting generalmente se refiere a código de programa escrito en JavaScript que es ejecutado por el navegador cuando se descarga una página, o en respuesta a un evento desencadenado por el usuario (W3C, 2015).

Las pruebas unitarias tienen ese nombre debido a que se prueba la funcionalidad de cada método o función; únicamente contemplando la lógica que debe realizar y excluyendo la convivencia con otras clases o sistemas.

El objetivo principal de las pruebas unitarias es asegurar que el código funciona realmente como se espera; no precisamente refleja lo que el cliente solicita, para ello existen otro tipo de pruebas como las de aceptación de usuario. Por lo tanto, las pruebas unitarias están destinadas para el aseguramiento del desarrollador y no para el usuario final.

Las pruebas de aceptación son principalmente para usuarios no técnicos como aprobadores, clientes, usuarios de sistema, etc.

Las pruebas de calidad de código sirven para garantizar que la calidad del código es realmente óptima y que la probabilidad de tener errores o bugs en la codificación es mínima.

PERT (Técnica de evaluación y revisión de programas) se desarrolló en la década de 1950 y se utilizó en forma amplia en la administración de proyectos militares de investigación y desarrollo. Una de sus principales características además de su capacidad

para identificar los programas y planes que se requieren para las tareas, es que se puede manejar las incertidumbres que existen e los pronósticos de tiempos para determinar diversas tareas.

CPM (Método de ruta crítica), fue desarrollado en forma independiente de PERT, pero está estrechamente relacionado con éste, se refiere básicamente a los intercambios entre el costo de un proyecto y su fecha de terminación. Sin embargo, no se hace distinción entre PERT o CPM, sino que se utiliza la designación colectiva entre PERT/CPM.

2.2 Marco metodológico

2.2.1 Metodología de la investigación

La metodología de la investigación se describe a continuación según las pautas de Arias (2006).

Nivel de investigación

La investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento.

El nivel de la investigación se refiere al grado de profundidad con que se estudia un objeto, en este caso el tipo de investigación según el nivel de profundidad es descriptiva ya que se puntualizó un problema específico y se le determinó una solución a partir de un modelo matemático definido con anterioridad. En esta investigación se estudiaron las características de los grafos y se realizó la descripción del problema de hamiltonicidad en grafos bipartitos balanceados.

2.2.2 Metodología del área aplicada

La metodología utilizada para el desarrollo de la aplicación para determinar la hamiltonicidad de los grafos bipartitos balanceados, se basó en CA/IO (Ciencia de la Administración e Investigación de Operaciones).

La CA/IO representa la aplicación de métodos cuantitativos para la toma de decisiones en todos los campos.

La investigación de operaciones y las ciencias de administración tienen múltiples y variados usos en todo tipo de organizaciones.

El proceso de solución de problemas CA/IO puede describirse en una estructura de seis etapas, como sigue:

1. Identificación, observación y planteamiento del problema

Es la primera etapa y comienza cuando quien toma las decisiones observa la realidad y percibe que el resultado que se desea no se está produciendo bajo las operaciones existentes. La descripción verbal del problema es una fase muy importante en esta etapa, porque es la base sobre el cual se planteará el modelo matemático.

2. Construcción del modelo

En esta etapa se incluye a quien construye el modelo y a quien toma las decisiones, en este punto se observa el problema con el objeto de identificar variables y relaciones clave, la observación del problema puede llevarse a cabo en forma colectiva o separada, sin embargo, es necesario realizar un enfoque unificado; por ello debe existir una interacción entre quien toma las decisiones y quien construye el modelo.

Esta etapa implica el desarrollo del modelo, pero antes de estructurar en forma matemática el problema es necesario examinar los factores identificados en la primera etapa. Puede resultar necesario hacer algunas consideraciones que limiten al problema real para que pueda resolverse. Con bastante frecuencia es necesario probar un planteamiento inicial del modelo para determinar las consideraciones que deben hacerse.

3. Generación de una solución

En esta tercera etapa ocurre el proceso de selección o desarrollo del algoritmo, existe un cierto grado de retroalimentación entre la segunda y tercera etapa, dado que debe tenerse la seguridad de que el problema planteado en la primera etapa satisface todas las

consideraciones que el algoritmo utiliza en esta etapa. Hay que tener cuidado con el término algoritmo porque un proceso de solución no se limita a un algoritmo, también se tienen los métodos heurísticos y simulación que se pueden utilizar para resolver determinados problemas.

4. Prueba y evaluación de una solución

En esta etapa se evalúa y se prueba el modelo adoptado o desarrollado en la etapa anterior, con el objeto de determinar si produce resultados útiles para el problema original. Pueden utilizarse diversos procedimientos para probar el modelo. En primer lugar, quien toma la decisión simplemente puede examinar los resultados y hacer algún juicio con respecto a cuán razonables son. En segundo lugar es posible adoptar un procedimiento de prueba a través del cual se utilicen situaciones historias previas como modelo base. Es decir, se puede introducir información proveniente de una decisión previa al modelo y comparar los resultados con lo que ocurrió en realidad. Sin importar si se utiliza uno de estos procesos de prueba, o ambos, para la evaluación del modelo, éste debe modificarse sino satisface las necesidades de quien toma las decisiones. Con bastante frecuencia el proceso de revisión implica añadir y eliminar variables, pero podría implicar volver al problema observado originalmente.

5. Implante

Esta quinta etapa es el implante del modelo validado. El implante comienza en el primer día del proyecto y no cuando el modelo se ha desarrollado y ya está operando. Los administradores deben participar el proyecto de CA/IO durante todo su desarrollo. El implante no significa que el modelo que se pone en práctica simplemente se entrega a los administradores y que, con esta acción, quien lo construyo se retira del proyecto. Quien construye el modelo debe trabajar con quien toma las decisiones para identificar en forma apropiada el problema en la primera etapa, obtener retroalimentación de quien toma las decisiones con respecto a la validez del modelo en la cuarta etapa, y trabajar con quien toma las decisiones para implantar y utilizar el modelo.

6. Evaluación

Esta última etapa es la evaluación y revisión del modelo. Dado que un modelo de CA/IO se utilice en forma repetida en el análisis de problemas de decisión, el modelo debe evaluarse en forma continua para determinar si los valores de los parámetros han cambiado y/o para ver si el modelo sigue satisfaciendo las metas de quien toma las decisiones, entonces debe considerarse una modificación del modelo.

En la Figura 6 se muestra una representación más detallada y práctica de este proceso. Las seis etapas del proceso se identifican con los cuadros de líneas punteadas numeradas.

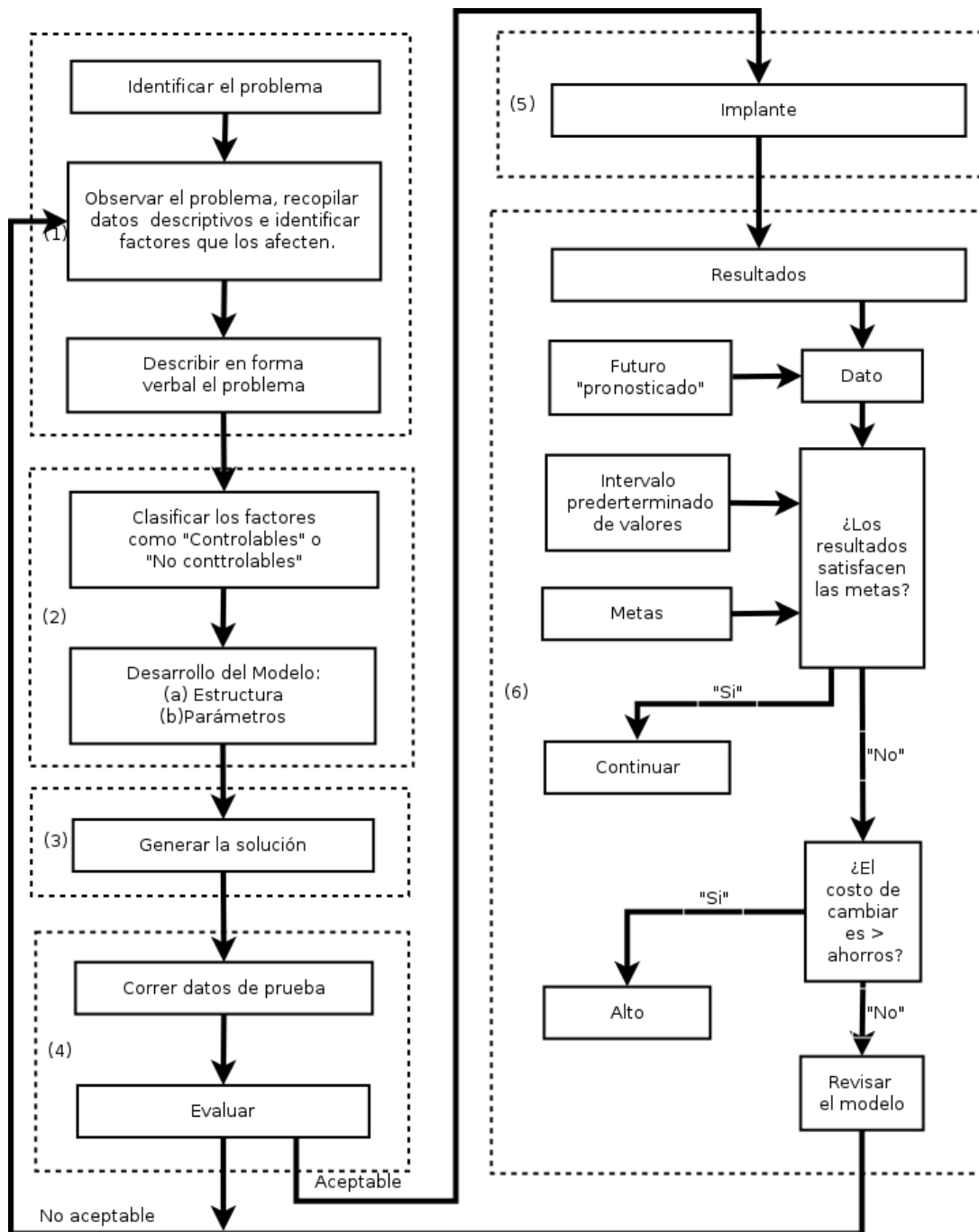


Figura 6. Proceso de solución de problemas de CA/IO

CAPÍTULO III: DESARROLLO

Este capítulo consistió en el desarrollo de la aplicación y en la ejecución de cada una de las fases de la metodología CA/IO.

Iteración 1

Identificación, observación y planteamiento del problema

En esta primera fase planteada por la metodología CA/IO se realizó un estudio al modelo matemático planteado.

Para entender el problema a fin de obtener los requisitos necesarios para el desarrollo de la aplicación y percibir el dominio del mismo, en primera instancia se desglosó el modelo matemático pudiendo definirse de forma más detallada:

Sea G un grafo bipartito balanceado de orden $2n$, es decir, un grafo con una bipartición en dos conjuntos de vértices independientes cada uno de ellos con n vértices, y mínimo grado, es decir, el menor número de lados incidentes a un vértice, al menos tres. Si para cada conjunto independiente balanceado de cuatro vértices, es decir, un conjunto independiente conteniendo dos vértices de cada lado de la bipartición, su número de vecinos es al menos $n + 1$, entonces G es hamiltoniano.

De esta forma se planteó el modelo en un árbol de decisión y se procede a exponer los pasos y la secuencia para realizar esta investigación. Ver Figura 7.

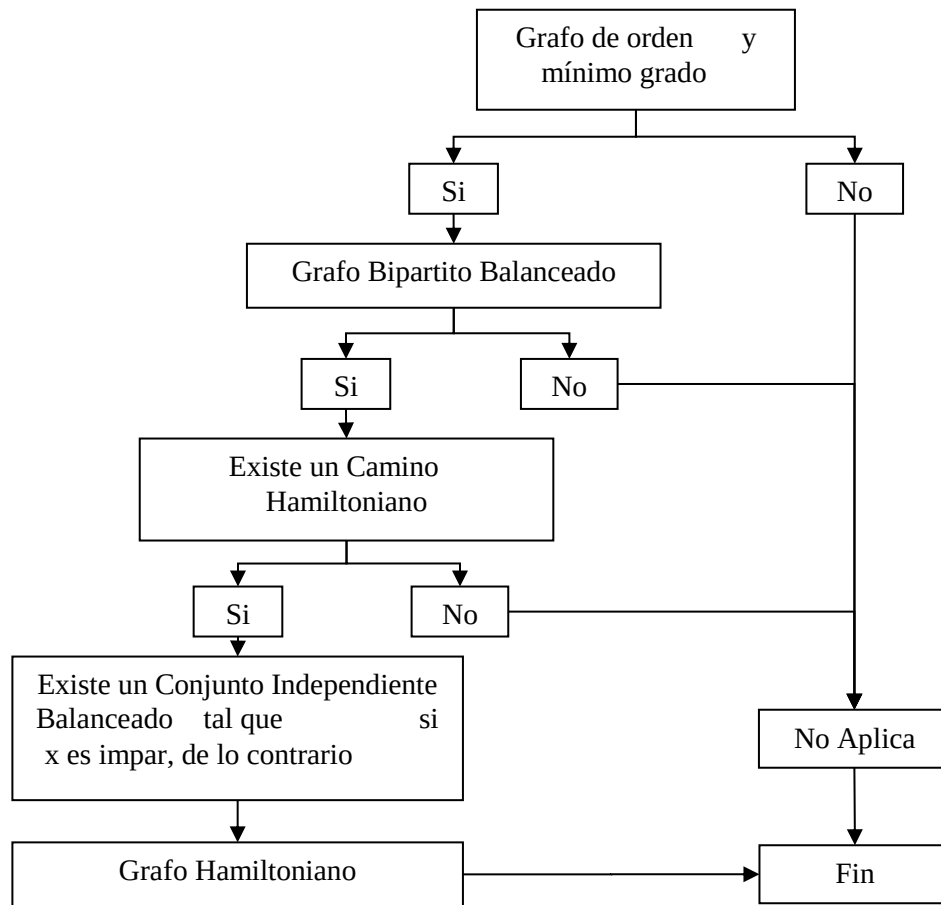


Figura 7. Árbol de decisión

Se obtuvo las variables del sistema, como son: el grado del grafo, el mínimo grado, la cantidad de vértices y la cantidad de aristas. Entre las restricciones tenemos que el orden del grafo debe ser par y el grafo debe ser bipartito balanceado.

Posteriormente se planificó el desarrollo del sistema, desglosando las fases de la metodología y las actividades que correspondían realizarse en cada una de ellas como se aprecia en la tabla 1. Una vez obtenida dichas actividades, se estudiaron y se asignaron a cada una de ellas un tiempo de duración pertinente para llevarse a cabo, seguidamente se determinó la precedencia de cada actividad. La tabla 2 muestra las actividades con su respectiva duración en semanas y precedencia.

Tabla 1. Lista de actividades para el diagrama PERT/CPM

Etiqueta	Actividad
A	Identificar el Problema
B	Estudio del modelo matemático.
C	Ingeniería de requisitos.
D	Identificación de variables para el sistema.
E	Diseño base de datos.
F	Diseño interfaz del sistema.
G	Calcular mínimo grado.
H	Verificar bipartición balanceada.
I	Calcular conjunto independiente de vértices.
J	Programar vistas de la interfaz del sistema.
K	Realizar pruebas a las soluciones realizadas.
L	Pruebas de integración.
M	Pruebas de usabilidad.
N	Pruebas de usuario.
O	Implante
P	Evaluación

Tabla 2. Lista de actividades para el diagrama PERT/CPM con duración en semanas y precedencia.

Etiqueta	Actividad	Duración	Precedencia
A	Identificar el Problema	1	-
B	Estudio del modelo matemático.	2	A
C	Ingeniería de requisitos.	4	A
D	Identificación de variables para el sistema.	2	B,C
E	Diseño base de datos.	2	D
F	Diseño interfaz del sistema.	5	D
G	Calcular mínimo grado.	4	E,F
H	Verificar bipartición balanceada.	5	G
I	Calcular conjunto independiente de vértices.	5	I
J	Programar vistas de la interfaz del sistema.	5	G,H,I
K	Realizar pruebas a las soluciones realizadas.	5	G
L	Pruebas de integración.	4	K
M	Pruebas de usabilidad.	2	L
N	Pruebas de usuario.	2	L
O	Implante	1	N
P	Evaluación	2	O

A continuación, se construyó el diagrama PERT/CPM. Normalmente se comienza de izquierda a derecha. Se emplean nodos para denotar inicio y fin de las actividades, líneas rectas horizontales para representar las actividades e identificadas con la etiqueta en la parte superior y la duración en la parte inferior. Las actividades ficticias se dibujan con líneas discontinuas, como se aprecia la Figura 8.

Se puede apreciar en la figura 8, que la duración del proyecto sería en 34 semanas, puesto que terminación de un proyecto requiere que se terminen todas las rutas de una red, la duración de la ruta más grande de una red es la ruta crítica.

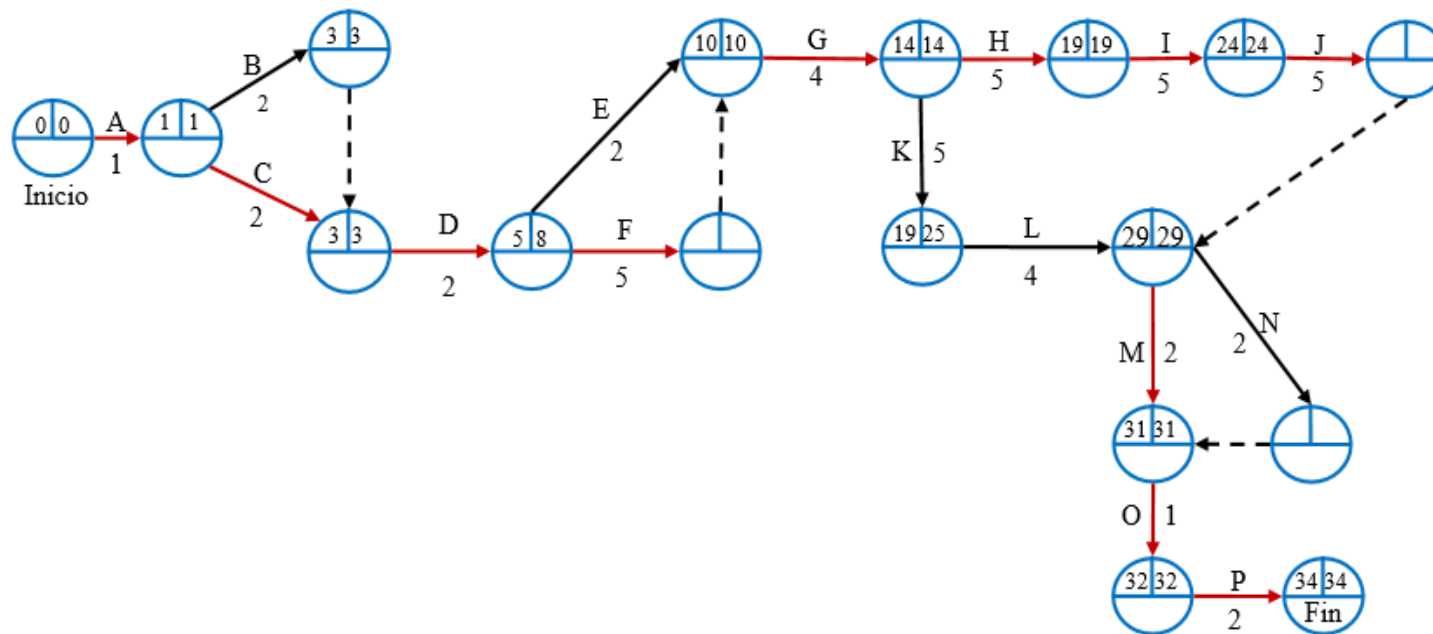


Figura 8. Diagrama PERT/CPM

Las actividades que se encuentran en la ruta crítica se les denominan actividades críticas.

Las actividades críticas para este proyecto son: A, C, D, F, G, H, I, J, M, O y P.

Desarrollo de requisitos

El proceso de ingeniería de requisitos se basó en expresar las necesidades de los usuarios que debían ser cumplidas por la aplicación. Se llevó a cabo una serie de reuniones con el Licenciado encargado del grupo de estudio del Departamento de Matemáticas con la intención de recaudar los requisitos desde el punto de vista de los usuarios.

Seguidamente se elaboró una lista contemplando las necesidades recaudadas mostradas en la Tabla 3. Más adelante, en un análisis más detallado se clasificó los requisitos obtenidos en funcionales y no funcionales. La Tabla 4 muestra la clasificación obtenida.

Tabla 3. Requisitos del sistema.

ID	Requisito
1	Ingresar el grafo.
2	Ingresar grafo en forma de conjuntos.
3	Ingresar grafo en forma gráfica
4	Calcular el mínimo grado del grafo.
5	Determinar bipartición balanceada.
6	Determinar conjuntos independientes de vértices.
7	Determinar hamiltonicidad.
8	Generar grafo
9	Mostrar en forma detallada los resultados del análisis.
10	Guardar grafos
11	Estar programado en lenguaje de código abierto y multiplataforma
12	Ser bajo ambiente de Web.
13	Desarrollar el sistema en lenguaje Python

Tabla 4. Requisitos del sistema y su clasificación.

ID	Requisito	Tipo de requisito
1	Ingresar el grafo.	Funcional
2	Ingresar grafo en forma de conjuntos.	Funcional
3	Ingresar grafo en forma gráfica	Funcional
4	Calcular el mínimo grado del grafo.	Funcional
5	Determinar bipartición balanceada.	Funcional
6	Determinar conjuntos independientes de vértices.	Funcional
7	Determinar hamiltonicidad.	Funcional
8	Generar grafo	Funcional
9	Mostrar en forma detallada los resultados del análisis.	Funcional
10	Guardar grafos	Funcional
11	Estar programado en lenguaje de código abierto y multiplataforma	No funcional
12	Ser bajo ambiente de Web.	No funcional
13	Desarrollar el sistema en lenguaje Python	No funcional

Una vez culminada la clasificación de los requisitos obtenidos, se procedió a elaborar el diagrama de casos de uso preliminar del sistema, que a continuación se muestra en la Figura 9 y 10 respectivamente.

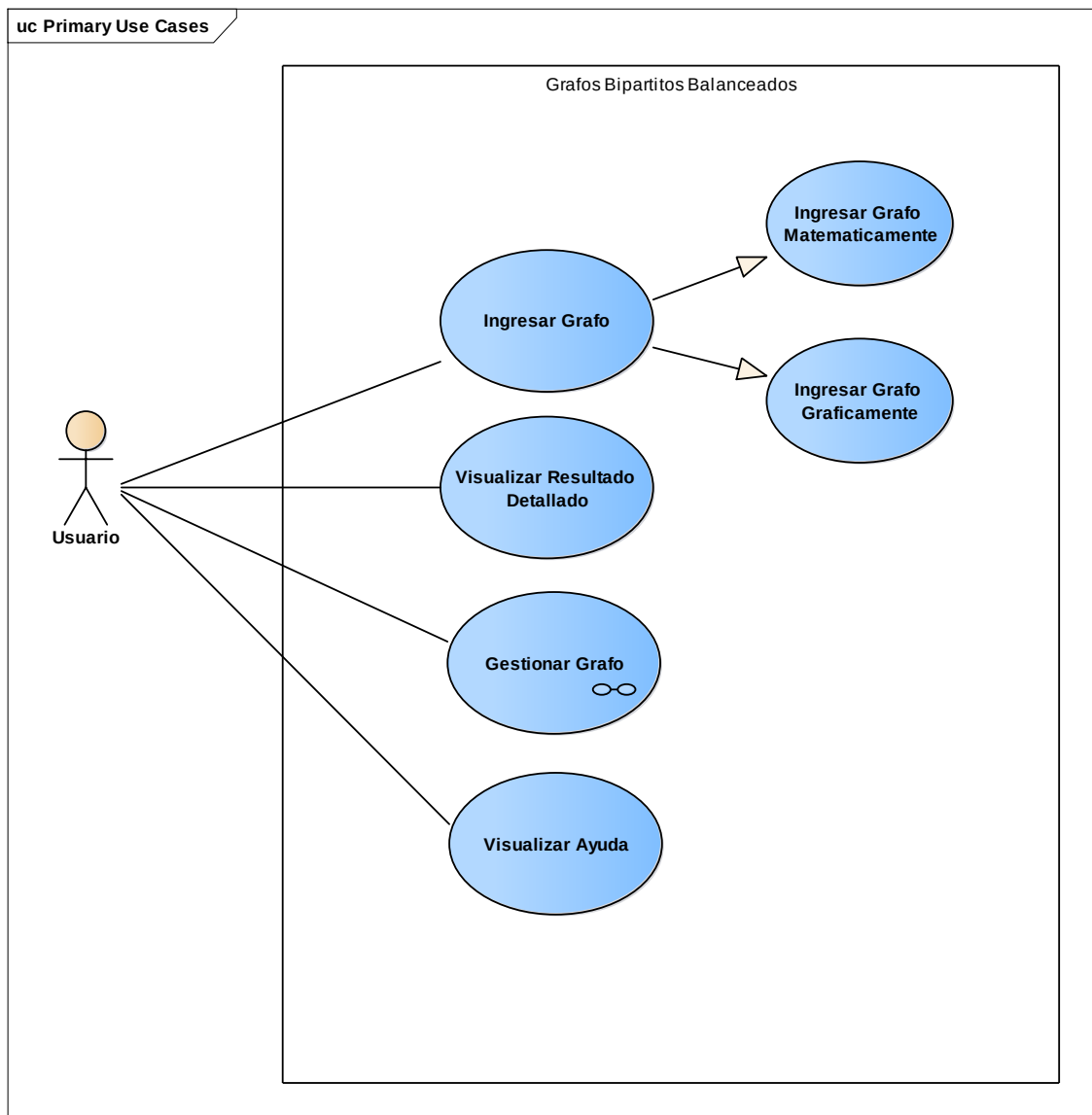


Figura 9. Diagrama de casos de uso del análisis de requisitos

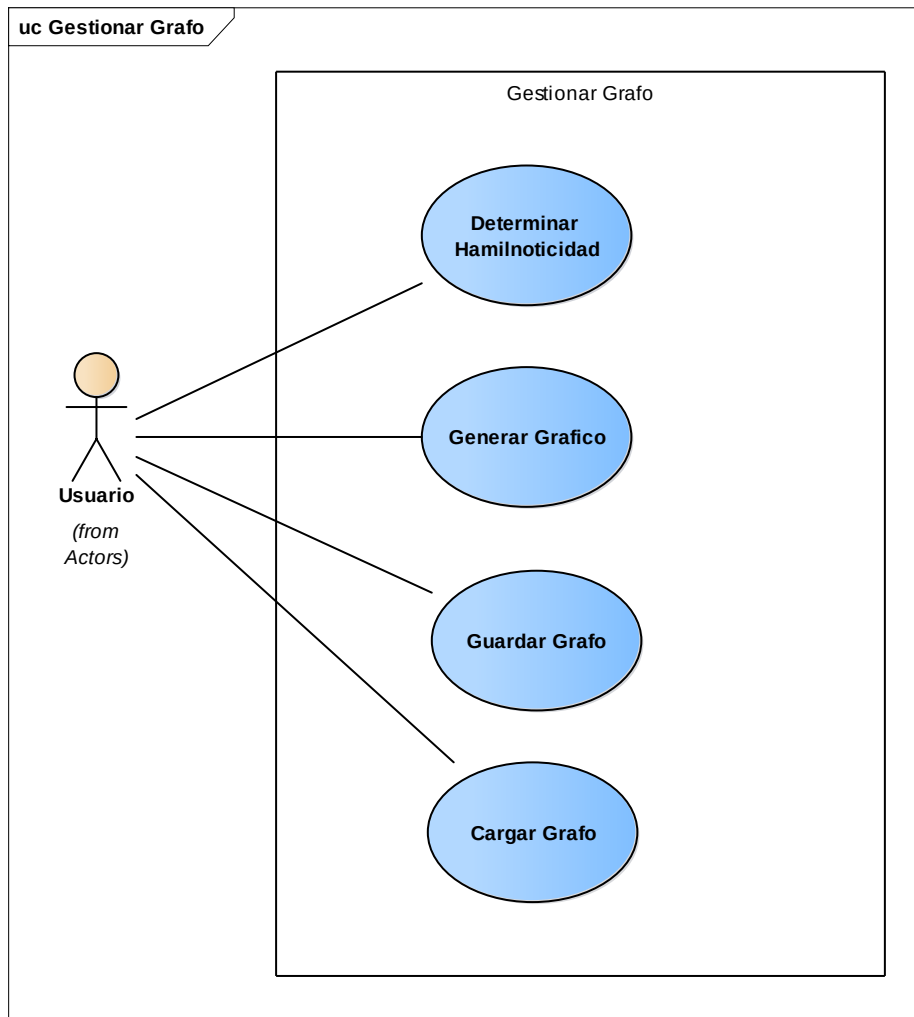


Figura 10. Diagrama de Casos de uso Gestionar Grafo

Construcción del modelo

En esta fase se establecieron las variables necesarias para la construcción del modelo utilizando como referencia el árbol de decisión planteado anteriormente, permitiendo así crear el diagrama de clases preliminar mostrado en la Figura 11.

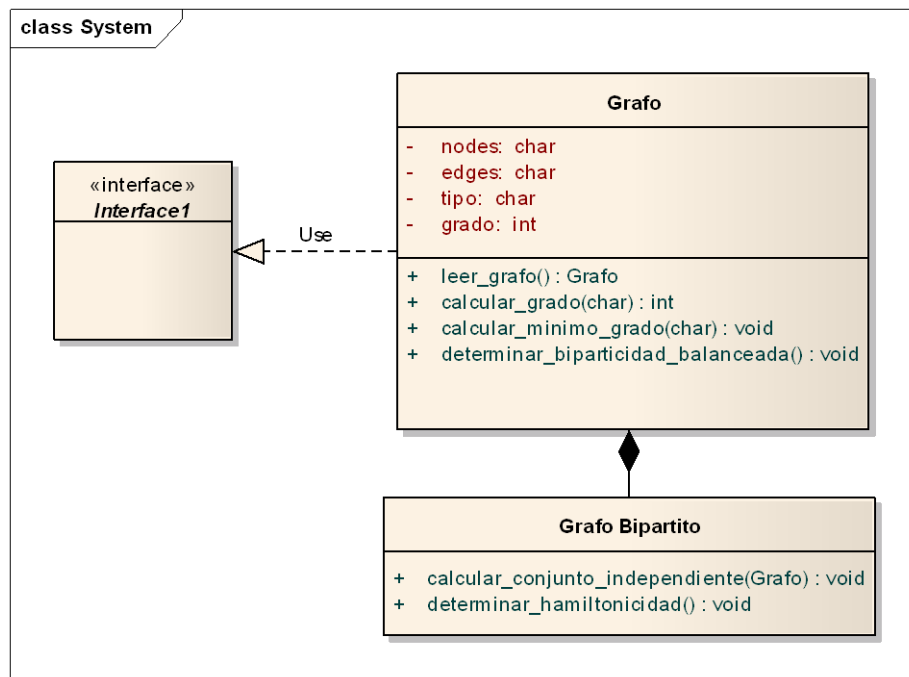


Figura 11. Diagrama de clases preliminar del modelo matemático

Generación de una solución

Esta primera iteración se abordó con la implementación de los grafos en el computador utilizando el lenguaje de programación Python y la librería NetworkX de Python. Se estructuró y codificó el diagrama de clase presentado anteriormente en la fase de construcción del modelo, desarrollando así el primer algoritmo que permite capturar el grafo a través de la consola de comando de Python, y de este modo se creó además el algoritmo que dibuja el grafo ingresado y lo muestra en pantalla.

Prueba y evaluación de una solución

Se realizaron pruebas unitarias a los algoritmos adoptados de la fase anterior verificando la lógica del mismo e ingresando grafos de cualquier número de cantidad de vértices y

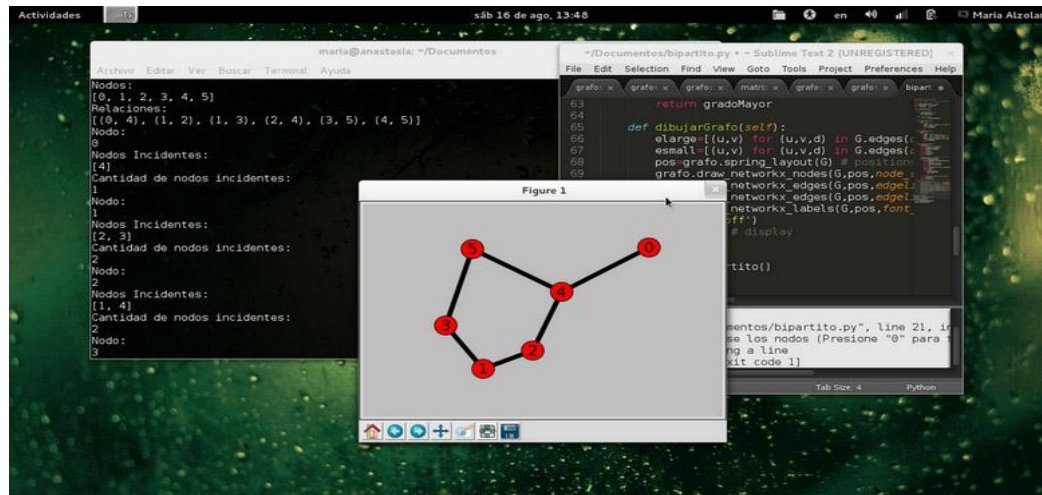


Figura 12. Prueba de grafo generado mediante la consola de comando de Python
nodos para verificar si la imagen del grafo generado corresponde a los datos ingresados por la consola. En la Figura 12 se muestra una prueba realizada a la imagen del grafo generado.

Iteración 2

Identificación, observación y planteamiento del problema

Basado en los requerimientos obtenidos en la iteración anterior se diseñó la interfaz

Conjunto de Vertices: Conjunto de Aristas:

☒ Grado ☐ Si Conjunto Independiente ☐ Si Camino Hamiltoniano ☐ Si Conjuntos Disjuntos

Figura 13. Diseño de interfaz preliminar para la aplicación

preliminar para la aplicación, mostrada en la Figura 13.

Generación de una solución

Continuando con el proceso de codificación en lenguaje Python se desarrolló el algoritmo para determinar el mínimo grado de un grafo dado, es decir, el menor número de lados incidentes a un vértice. A continuación, se muestra el algoritmo para determinar el mínimo grado de un grafo:

Algoritmo 1: Determinar el mínimo grado de un grafo

1. Inicio
2. Entrada: nodos y lados del grafo
3. Salida: mínimo grado del grafo
4. Var var, cont , miniGrado
5. Para cada nodo x de nodosGrafo hacer
 - 5.1 $\text{cont} \leftarrow$ Calcular la cantidad de vecinos de cada nodo x
 - 5.2 si ($\text{miniGrado}=0$ y $\text{cont}>\text{miniGrado}$) hacer
 - miniGrado=cont
 - 5.3 si ($\text{miniGrado}\neq 0$ y $\text{cont}<\text{miniGrado}$) hacer
 - miniGrado $\leftarrow$ cont
6. Devolver miniGrado
7. Fin

Implementación del algoritmo en lenguaje Python:

```
def minimoGrado(self):
    for x in G.nodes():
        var=G.neighbors(x)
        cont=len(var)
        if self.miniGrado ==0 and cont>self.miniGrado:
            self.miniGrado = cont
            Pass
        if self.miniGrado!=0 and cont<self.miniGrado:
            self.miniGrado = cont
            Pass
    Pass
    return self.miniGrado
```

Para determinar la bipartitidad de un grafo es necesario dividir el mismo en dos conjuntos disjuntos A y B tal que cada vértice del conjunto A este relacionado con un vértice del conjunto B , para realizar esto es necesario tener una combinación de los posibles conjuntos que se pueden formar con el grafo y hacer las respectivas validaciones, lo que se convierte en una combinación de vértices. A continuación, se puede apreciar el Algoritmo 2 que se encarga de calcular las combinaciones.

Algoritmo 2: Determinar combinaciones

1. Inicio
2. Entrada: nodos del grafo
3. Salida: combinación de nodos
4. Var combinar, lista, listanueva, mitad, conjunto, new_comb
5. Para cada elemento x de combinar hacer
 - tamano $\leftarrow$ calcular el tamaño x
 - 5.1 si(tamano<mitad) hacer
 - 5.1.1 pto $\leftarrow$ Hacer que el nuevo elemento agregado de 'lista', sea el siguiente en posición del último carácter del elemento 'combinar'.
 - 5.1.2 Para cada elemento de 'lista' desde el elemento 'lista' que hace parte del último carácter del elemento de 'combinar' hasta el último elemento de 'lista' hacer
 - 5.1.2.1 Si el elemento de 'lista' no está en el elemento 'combinar' y además el elemento 'lista' no es el último carácter del elemento 'combinar' hacer
 - 5.1.2.1.1 new_comb $\leftarrow$ agrego un nuevo elemento 'lista' a la combinación
 - 5.1.3 listanueva $\leftarrow$ new_comb
6. Devolver listanueva

Implementación del algoritmo en lenguaje Python:

```
def combinacion(self,combinar,lista,listanueva,mitad,conjunto):
    new_comb=[]
    for x in combinar:
        tamano = len(x)
        if tamano<mitad:
            pto=lista.index(x[len(x)-1])
            for j in range(pto,len(lista)):
                aux=[]
```

```

        if lista[j] not in x and lista[len(lista)-1] not in x:
            aux.append(x+lista[j])
            new_comb.append(x+lista[j])
            conjunto.append(x+', '+lista[j])
        listanueva=new_comb
    Pass
return listanueva

```

Una vez obtenida las combinaciones de los conjuntos de vértices del grafo, se procede a verificar y encontrar dos conjuntos disjuntos para así concretar si el grafo es bipartito, ver algoritmo 4.

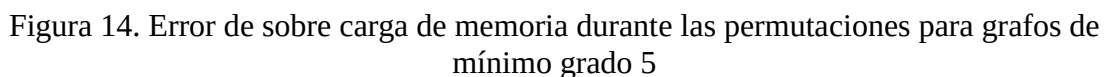
Algoritmo 4: Conjunto disjunto

```

1  Inicio
2  Entrada: nodos del grafo, conjunto
3  Salida: conjunto disjunto del grafo
4  Var lados, conjunto
5  conjuntos ← conjunto
6  conjunto_disjunto ← []
7  Para subconjunto x del conjunto hacer
    7.1  tamaño←calcular tamaño del subconjunto x
    7.2  cont← 0
    7.3  comb_final← x
    7.4  conjunto←[]
    7.5  encontrado←false
    7.6  Mientras cont<1 hacer
        comb_final=combinacion(comb_final,x,listanueva,2,conjunto)
        cont = cont + 1
    7.7  Para cada sub-conjunto i del comb_final hacer
        7.7.1  Para cada elemento k de lados hacer
            7.7.1.1  Si (i[0]=k[0] and i[1]=k[1]) hacer
                7.7.1.1.1  encontrado ← true
                7.7.1.1.2  k←lado
    7.8  Si (encontrado← false) hacer
        7.8.1  conjunto_disjunto← agregar subconjunto x
8  Devolver conjunto_disjunto

```

Ingresando grafos de orden par y de grado mínimo 3 el algoritmo obtuvo resultados satisfactorios en cuanto a la exactitud de los resultados con un tiempo de duración de 3.002512 segundos. Sin embargo, al ejecutar el algoritmo con grafos de orden par y de grado mínimo 5, se apreció un gran consumo de recurso del computador, en cuanto a memoria se refiere; esto sin duda a la gran cantidad de permutaciones a realizarse, ocasionando el colapso del algoritmo y cierre forzoso por el sistema operativo al consumirse toda la memoria principal, tal como se aprecia en la figura 14.



Implante

En esta etapa se comprobó la veracidad de los resultados arrojados por el algoritmo para grafos de mínimo grado 3. En primera instancia se propuso la revisión absoluta de la secuencia y comportamiento del algoritmo, para comprobar si podrían hacerse mejoras en el mismo para aumentar su productividad, debido al consumo de memoria que absorbía su ejecución, principalmente las combinaciones de los vértices se capturaban en una variable del sistema, debido a la gran cantidad de combinaciones dicha variable consumía la capacidad de memoria RAM del computador y ocasionada el colapso del sistema.

Iteración 3

Generación de una solución

Atendiendo las consideraciones de la iteración anterior, se realizó un estudio exhaustivo del algoritmo con la finalidad de depurar el mismo y evitar hacer uso de memoria innecesaria. Se realizaron mejoras en la implementación del algoritmo, evitando la asignación de los resultados de las permutaciones en variables ya que al ser muchas se consumían toda la memoria RAM del computador y como resultado se redujo el tiempo de ejecución del mismo, menos consumo de memoria y procesador, permitiendo ahora cálculos a grafos de grado mínimo 5.

Una vez depurado el código para la comprobación de un grafo bipartito balanceado, el siguiente paso fue el desarrollo del código que se encarga de extraer todos los posibles conjuntos independientes balanceados de vértices del grafo, ver algoritmo 5.

Algoritmo 5: Conjunto independiente

1. Inicio
2. Entrada: conjuntoA, conjuntoB, lados
3. minimoGrado ← tomar el mínimo grado del grafo
4. cant ← 0
5. si minimoGrado es par hacer

```

    5.1 cant ← minimoGrado
6. sino hacer
    6.1 cant ← minimoGrado + 1
7. cant ← cant/2
8. tamano_l ← tamaño del conjuntoA
9. tamano_x ← tamaño del conjuntoB
10. cont ← 0
11. comb_a←conjuntoA
12. comb_b←conjuntoB
13. encontrado ← False
14. mientras cont<cant-1 hacer
    comb_a←combinacion(comb_a,conjuntoA,lista_a,cant,conjunto_a)
    comb_b←combinacion(comb_b,conjuntoB,lista_b,cant,conjunto_b)
    cont ← cont +1
15. para cada elemento v de comb_a hacer
    15.1 vecinos ← []
    15.2 para cada elemento f de comb_b hacer
        15.2.1 para cada elemento i de v hacer
            15.2.1.1 var←tomar los vecinos de i
            15.2.1.2 para cada elemento k de var hacer
                15.2.1.2.1 si k se encuentra en vecinos hacer
                    vecinos ← agregar k a vecinos
        15.2.2 para cada elemento i de f hacer
            15.2.2.1 var←tomar los vecinos de i
            15.2.2.2 para cada elemento k de var hacer
                15.2.2.2.1 si k se encuentra en vecinos hacer
                    vecinos ← agregar k a vecinos
        15.2.3 Hamiltoniano ← True
        15.2.4 s ← calcular tamaño de vecinos
        15.2.5 cantv ←calcular tamano de v
    15.2.6 cantf ←calcular tamano de f
    15.2.7 N ← cantv+cantf
    15.2.8 si (s<N+1) hacer
        Hamiltoniano ← False
        Retornar False
16. si (Hamiltoniano=True) hacer
    Retornar True

```

Implementación del algoritmo en lenguaje Python

```
def conjuntosIndependiente(self,conjuntoA,conjuntoB,edges):
    minimoGrado=self.miniGrado
    cant = 0
    if minimoGrado%2==0:
        cant = minimoGrado
    else:
        cant = minimoGrado + 1
    cant = cant/2
    tamano_l = len(conjuntoA)
    tamano_x = len(conjuntoB)
    cont = 0
    comb_a=conjuntoA
    comb_b=conjuntoB
    lista_a=[]
    conjunto_a=[]
    lista_b=[]
    conjunto_b=[]
    encontrado = False
    while cont<cant-1:
        comb_a=self.combinacion(comb_a,conjuntoA,lista_a,cant,conjunto_a)
        comb_b=self.combinacion(comb_b,conjuntoB,lista_b,cant,conjunto_b)
        cont = cont + 1
        Pass
    for v in comb_a:
        vecinos = []
        for f in comb_b:
            for i in v:
                var=G.neighbors(i)
                for k in var:
                    if k not in vecinos:
                        vecinos.append(k)
                    Pass
                Pass
            Pass
        Pass
        for i in f:
            var=G.neighbors(i)
            for k in var:
                if k not in vecinos:
```

```

                                vecinos.append(k)
                                Pass
                            Pass
                        Pass
                    Pass
                Hamiltoniano = True
                s = len(vecinos)
                cantv = len (v)
                cantf = len (f)
                N = cantv+cantf
                if s<N+1:
                    Hamiltoniano = False
                    return False
                    pass
            Pass
        if Hamiltoniano==True:
            return True
        Pass

```

Para esta iteración se tienen todas las herramientas para verificar la bipartitividad de un grafo dado en función de su mínimo grado y número de vecinos.

En este sentido se procedió a la unión de cada código y crear así un programa que contenga todos los módulos enlazados. Al mismo tiempo basándonos en el diseño preliminar de la interfaz para la captura del grafo se construyó la vista de dicha interfaz, utilizando Django como framework de desarrollo de aplicaciones web de Python, mostrada en la figura 15.

Se hizo necesaria la creación de un intérprete para la captura de los grafos, y así poder ingresarlos en forma matemática, a través de la interfaz desarrollada. En la Figura 16 se aprecia la forma de ingresar el grafo a través de la interfaz.

GRAFO BIPARTITOS

Conjunto de Vértices:

{A,B,C,D,E,F,G,H,I,J,K,L,M,N}

Conjunto de Aristas:

((A,C)(A,D)(A,E)(A,F)(A,G)(B,C)(B,D)(B,E)(B,F)(B,G)(M,L)(M,K)(M,J)(M,H)(M,I)(N,L)(N,K))

Generar

Minimo Grado:

Conjunto Independiente

Camino Hamiltoniano

Conjuntos Disjuntos

Informe Detallado

Resultados

Minimo Grado	Hamiltoniano
3	Si
Conjuntos Disjuntos	
Conjunto A	Conjunto B
CEDGFMN	CEDGFMN

Figura 15. Vista de interfaz para el módulo de los grafos

GRAFO BIPARTITOS

Conjunto de Vértices:

{A,B,C,D,E,F}

Conjunto de Aristas:

((A,D)(A,E)(B,D)(B,E)(C,D)(C,E)(C,F)(F,B)(F,A))

Generar

Figura 16. Intérprete de grafos

Prueba y evaluación de una solución

Se realizaron pruebas al intérprete desarrollado en la etapa anterior capturando grafos desde la interfaz del sistema, dicha captura se realizó siguiendo los patrones de expresión de grafos en forma de conjuntos como se conoce matemáticamente. En las figuras 17 y 18 se puede ver una prueba realizada ingresando datos erróneos.

40

GRAFO BIPARTITOS

Conjunto de Vértices:

Conjunto de Aristas:

Generar

Expresión Incorrecta

Figura 18. Resultado del intérprete al ingresar vértices erróneos

GRAFO BIPARTITOS

Conjunto de Vértices:

Conjunto de Aristas:

Generar

Expresión Incorrecta

Expresión Incorrecta

Figura 17. Resultado del intérprete al ingresar aristas y vértices erróneos

Al mismo tiempo se efectuaron las pruebas para verificar el correcto funcionamiento de la integración de todos los algoritmos.

Iteración 4

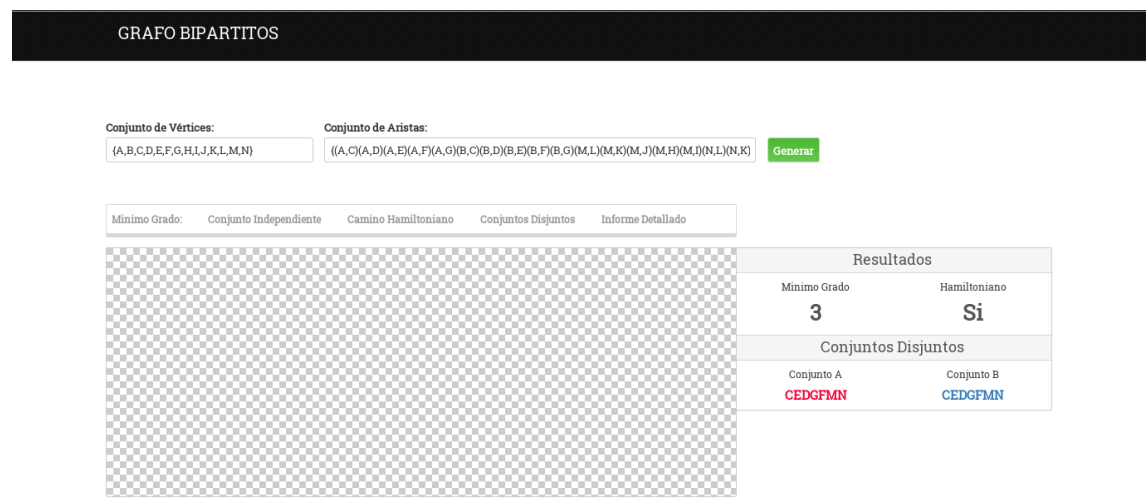
Generación de una solución

En esta iteración se reformaron y corrigieron detalles encontrados en la interacción anterior. Se refino la interfaz y la forma en cómo se mostraban los resultados.

Implante

En esta etapa del desarrollo de la aplicación, se utilizó para estudiar grafos de diferentes grados, arrojando información como el mínimo grado del mismo, si cumple con las características que plantea el modelo matemático, muestra el conjunto disjunto de vértices que satisfacen dichas propiedades y muestra la gráfica del grafo.

En la Figura 19 se puede apreciar la aplicación arrojando los resultados del procesamiento de un grafo ingresado. Así mismo en la Figura 20 se puede ver la gráfica del grafo ingresado a través del intérprete.



Evaluación

El modelo se evaluó para determinar si produce resultados útiles para el problema original, se comprobó la veracidad de los resultados arrojados por la aplicación. Se hicieron pruebas con grafos simples y de orden par, desde mínimo grado 2 hasta mínimo grado 8.

La figura 21 muestra el resultado obtenido al ingresar un grafo de orden par y de mínimo

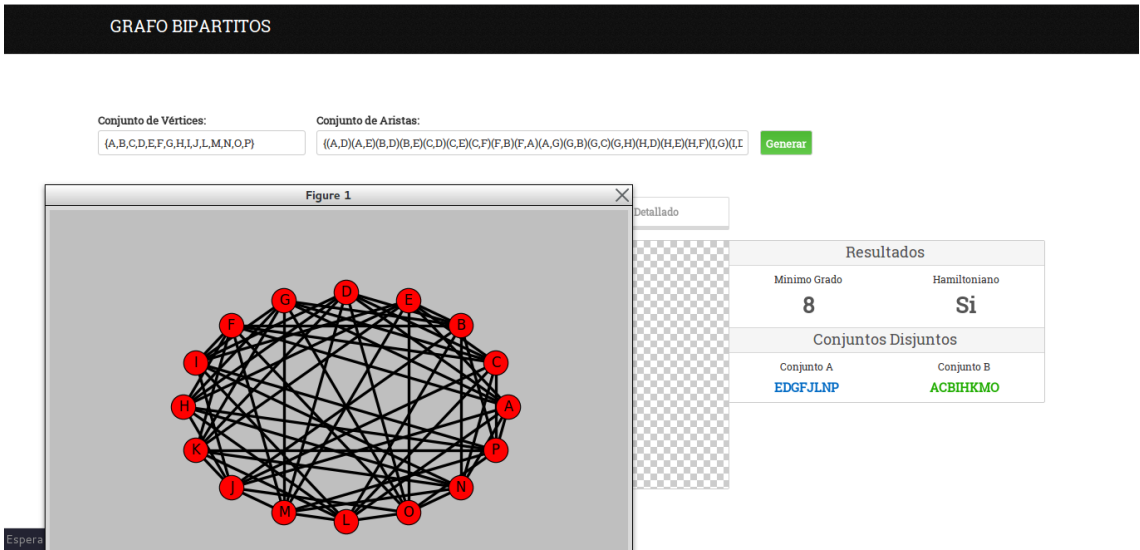


Figura 21. Resultados de la aplicación al graficar un grafo grado 8.

CONCLUSIONES

Una vez culminada la ejecución de cada una de las fases de la metodología CA/IO en el desarrollo de la aplicación que permite determinar la hamiltonicidad de un grafo bipartito balanceado en función de su mínimo grado y número de vecinos, es posible plantear las siguientes consideraciones finales:

El empleo de la metodología CA/IO garantizó la calidad y buen desarrollo de la aplicación ya que las fases se adaptaron a los requerimientos establecidos durante el proceso de desarrollo. Esta metodología permitió entre otros aspectos, el diseño de una interfaz simple, fácil de usar y acorde a las necesidades que correspondían satisfacerse.

Dada la flexibilidad y adaptabilidad que tiene PERT/CPM a cualquier proyecto, fue posible su utilización para la planificación de esta investigación.

UML se ajustó de forma acertada a la investigación, ya que ayudó en el proceso de recolección de información y al entendimiento e interpretación de la misma. Se utilizó el diagrama de casos de uso para expresar la funcionabilidad de la aplicación y el diagrama de clases para enunciar las clases necesarias.

El uso del lenguaje de programación Python permitió alcanzar una mayor productividad, puesto que simplifica mucho la programación al proponer un patrón a seguir; adicionalmente, es un lenguaje elegante, flexible; la programación es sana y productiva, muy sencillo de aprender, intuitivo y además muy leíble.

La aplicación desarrollada agiliza el proceso de determinar la hamiltonicidad de un grafo bipartito en función del mínimo grado y su número de vecinos. Además disminuye el tiempo requerido para corroborar si un grafo es hamiltoniano.

RECOMENDACIONES

Una vez realizada la investigación y planteadas las consideraciones finales, nos permitimos sugerir las siguientes recomendaciones:

Elaborar un plan para la implementación del sistema desarrollado, donde se contemple y especifique el adiestramiento para los usuarios finales así como la distribución de la aplicación en los servidores destinados como plataforma tecnológica.

Integrar al sistema módulos que contengan algoritmos y procesos del área de teoría grafos con el fin de crear una plataforma de investigación más completa que esté a la disposición de investigadores del área.

Incentivar a investigadores del área de estudio a implementar los algoritmos de este sistema en forma paralela para aprovechar los recursos del computador, reducir el tiempo de respuesta del mismo y ampliar su capacidad para permitir grafos con mayor cantidad de vértices.

Implementar métodos que permitan mejorar los tiempos computacionales de respuesta.

Desarrollar un módulo que permita el análisis de resultados.

BIBLIOGRAFÍA

Alcalá, Y.; Brito, D. y Marin, L. (2013). *The Hamiltonicity of Balanced Bipartite Graphs Involving Balanced Independent Set*. International Mathematical Forum, Vol 8, No 2 1353-1358.

Arias, F. (2006). *El proyecto de investigación* (6ta ed.). Caracas: Episteme.

Amar, D.; Brandt, S.; Brito, D. y Ordaz, O. (1998). *Neighborhood conditions for balanced independent sets in bipartite graphs*. Discrete Mathematics, 181, 31-36.

Bruto, D. y Lárez, G. (2007). *Neighborhood conditions for balanced bipartite graphs to be Hamiltonian*, International Journal of Pure and Applied Mathematics, 34, 509-512.

Booch, G., Rumbaugh, J. y Jacobson, I. 2004. *El Lenguaje Unificado de Modelado UML*. Addison Wesley Iberoamericana, Madrid.

Caicedo A.; Wagner, G. y Méndez, R. (2010). *Introducción a la teoría de grafos* [Libro en Línea]. Consultado el 06 de Noviembre de 2013 en: <http://books.google.co.ve/books?id=3hH11r7j1tcC&printsec=frontcover&hl=es#v=onepage&q&f=false>

Coss, R. (2003). *Simulación un enfoque práctico* [Libro en Línea]. Consultado el 22 de Febrero de 2014 en: https://books.google.co.ve/books?id=iY6dI3E0FNUC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Davis, K y McKeown P. (1984). *Modelos cuantitativos para administración*. California: Iberoamérica.

García, Y. (2012). *Aplicación para determinar si una sucesión de números es bipartita gráfica* (Tesis de Pregrado). Universidad de Oriente, Venezuela.

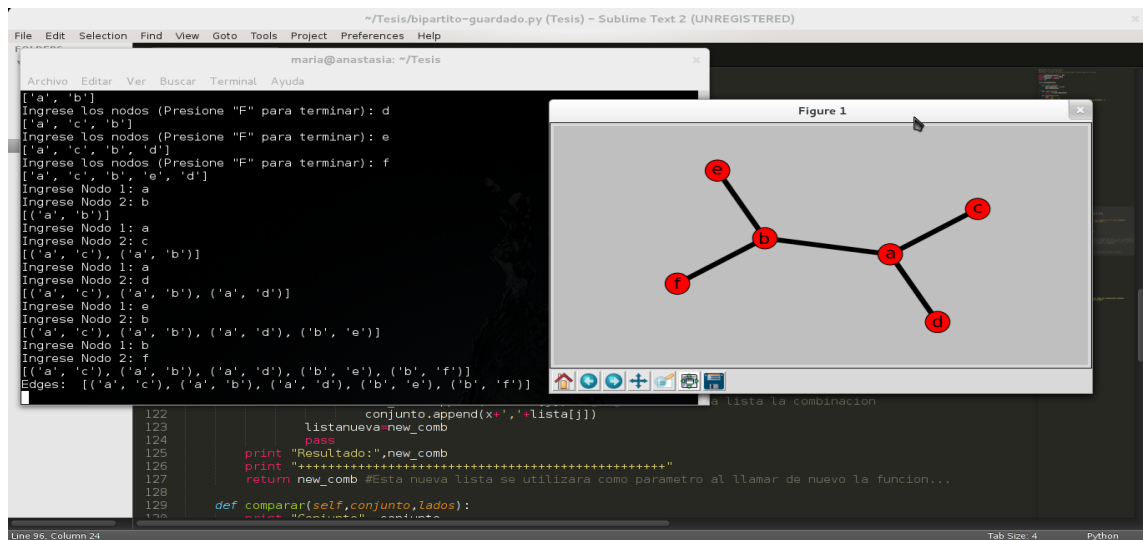
Giudici, R y Bris, A. (1997). *Introducción a la Teoría de Grafos* [Libro en Línea]. Consultado el 17 de Febrero de 2014 en: http://books.google.co.ve/books?id=0MqeE30_C6cC&printsec=frontcover&hl=es#v=onepage&q&f=false

W3C. 2010. *El W3C de la A a la Z*. Sitio Web: <http://www.w3c.es/Divulgacion/a-z/>

APÉNDICES

INDICE

Apéndice A: Resultados de las pruebas unitarias de los algoritmos	49
Apéndice B: Resultados de pruebas de la aplicación	51
Apéndice C: Manuales de la aplicación web	56
Apéndice D: Manuales de la aplicación	57



Apéndice A: Resultados de las pruebas unitarias de los algoritmos

Figura A.1. Resultado de la prueba del algoritmo para graficar un grafo

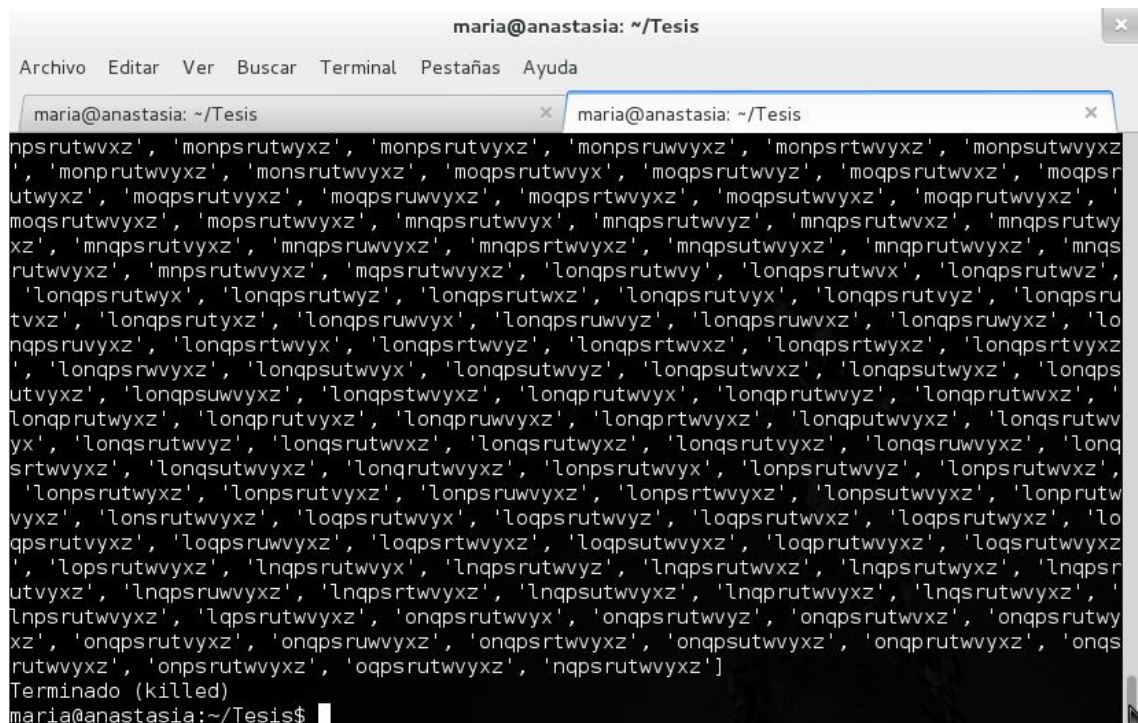


Figura A.2. Error de sobre carga de memoria durante las permutaciones para grafos de mínimo grado 5

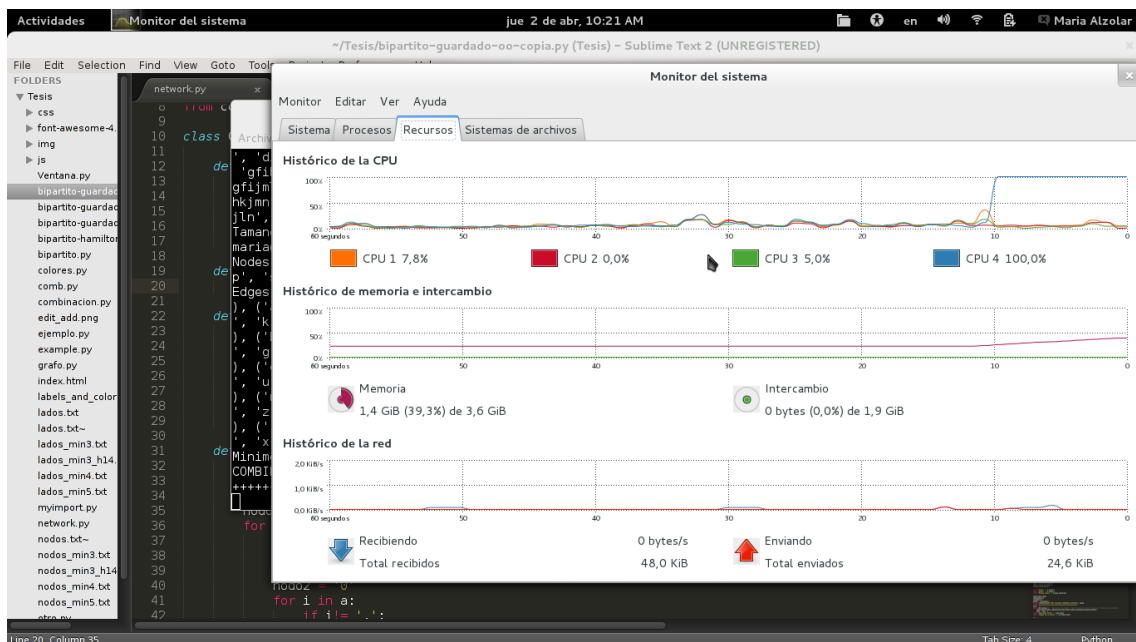


Figura A.3. Monitor del sistema al inicio del algoritmo para grafos de m3nimo grado 5

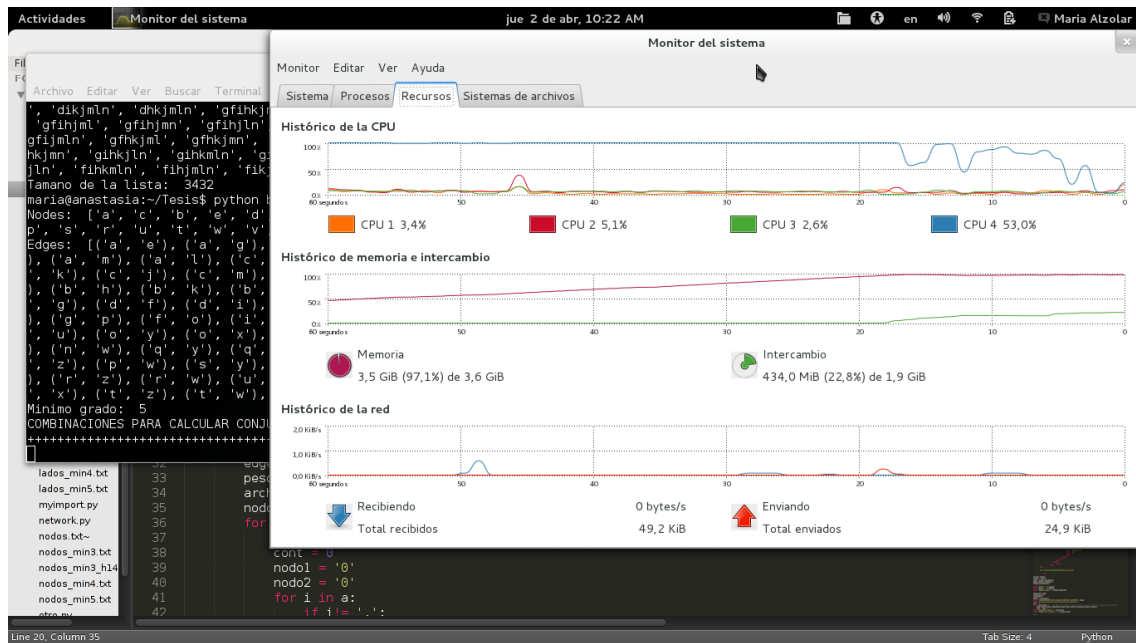


Figura A.4. Monitor del sistema a un minuto despu3s de la ejecuci3n del algoritmo para grafos de m3nimo grado 5

GRAFO BIPARTITOS

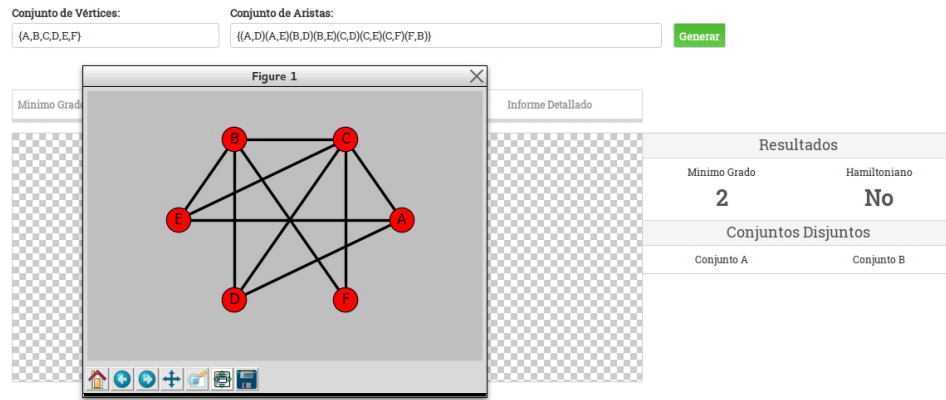


Figura B.3. Resultado de prueba con grafo de grado par, orden 6 y mínimo grado 2

GRAFO BIPARTITOS

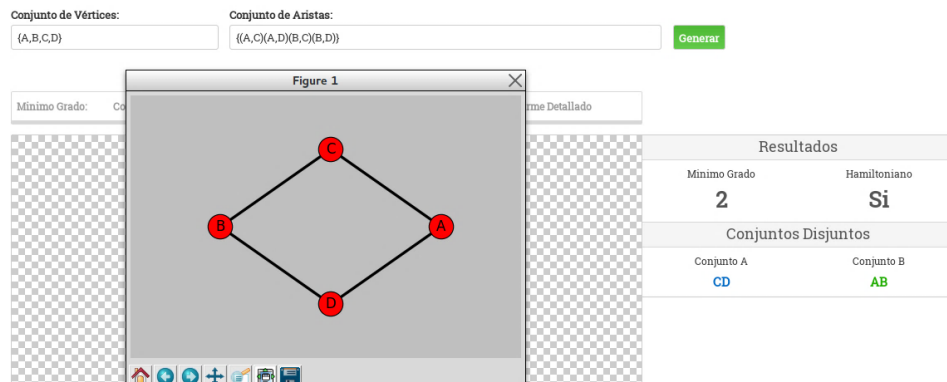


Figura B.2. Resultado de prueba con grafo bipartito de grado par y mínimo grado 2

GRAFO BIPARTITOS

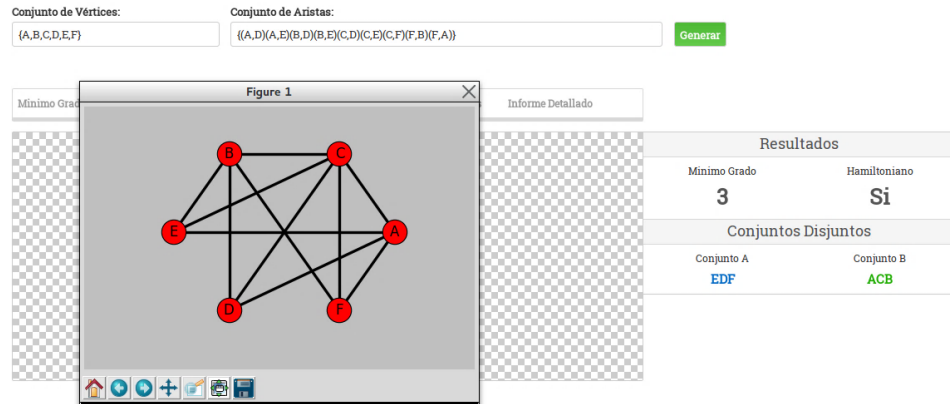


Figura B.4. Resultado de prueba con grafo de grado par, orden 6 y mínimo grado 3

GRAFO BIPARTITOS

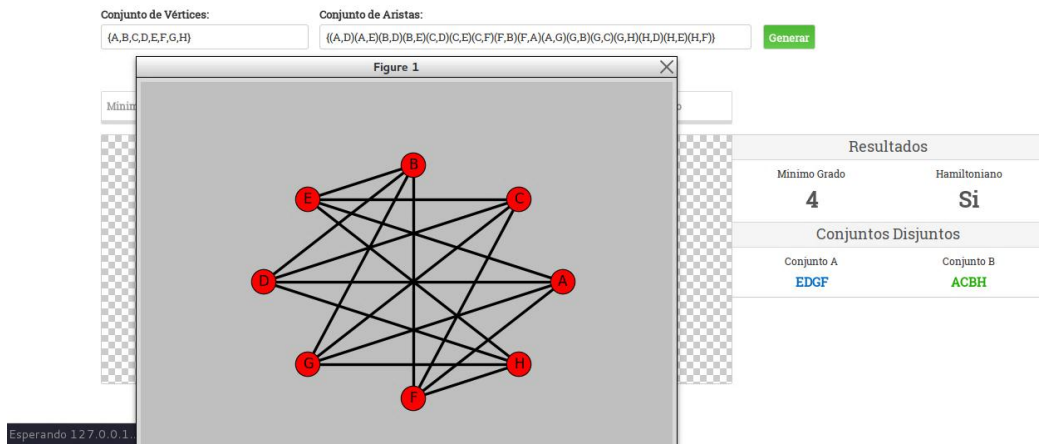


Figura B. 5. Resultado de prueba con grafo de orden 8 y mínimo grado 4

GRAFO BIPARTITOS

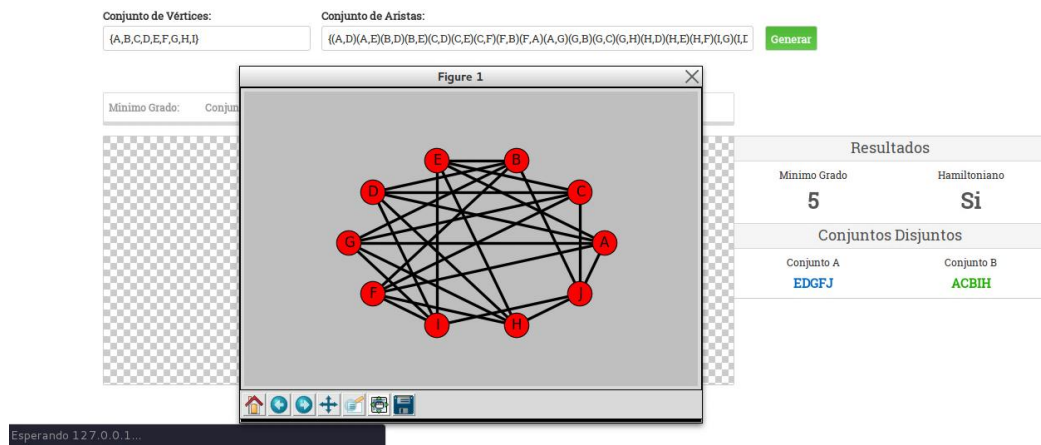


Figura B.6. Resultado de prueba con grafo de orden 10 y mínimo grado 5

GRAFO BIPARTITOS

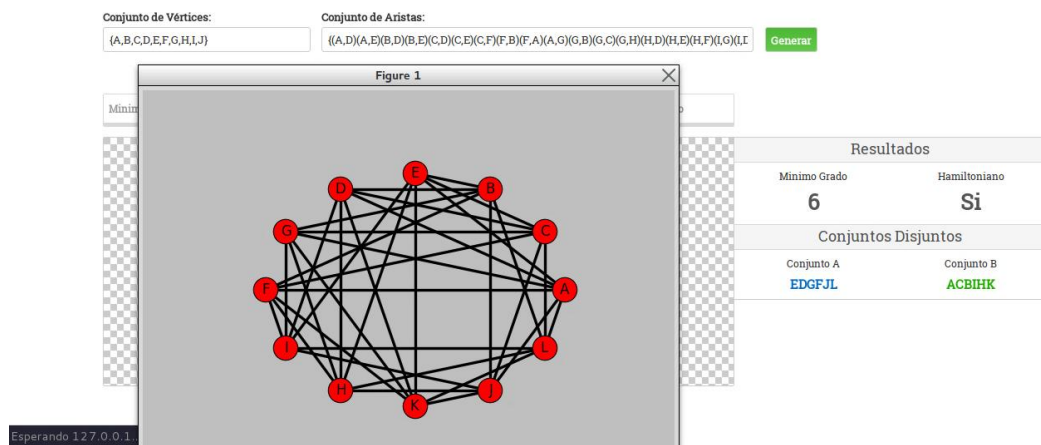


Figura B.7. Resultado de prueba con grafo de orden 12 y mínimo grado 6

GRAFO BIPARTITOS

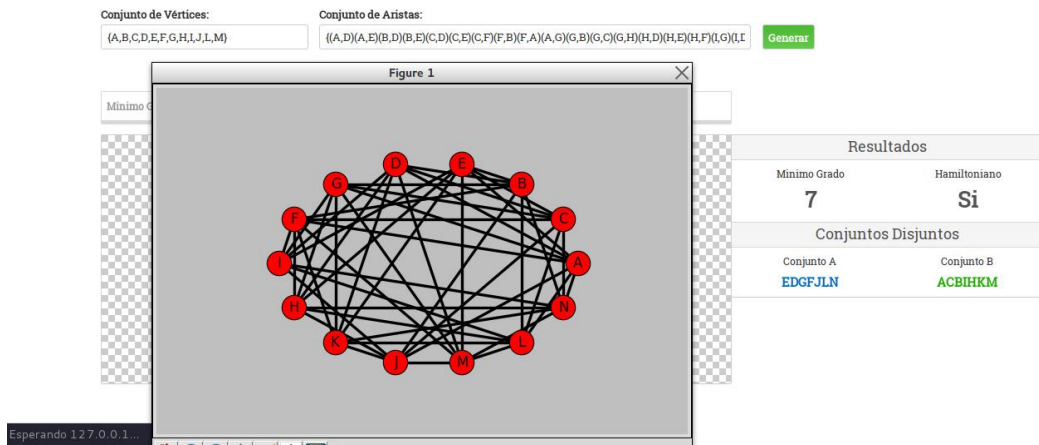


Figura B.8. Resultado de prueba con grafo de orden 14 y mínimo grado 7

GRAFO BIPARTITOS

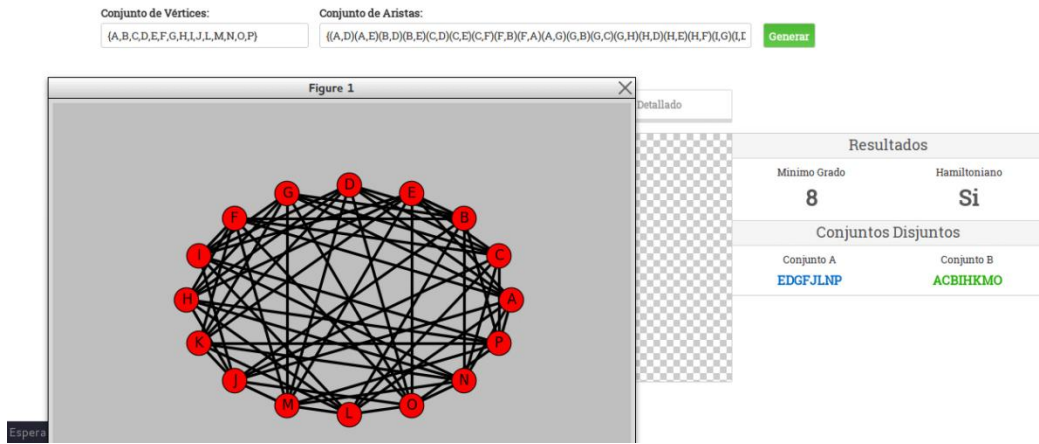


Figura B.9. Resultado de prueba de grafo de orden 16 y mínimo grado 8

Apéndice C: Manuales de la aplicación web

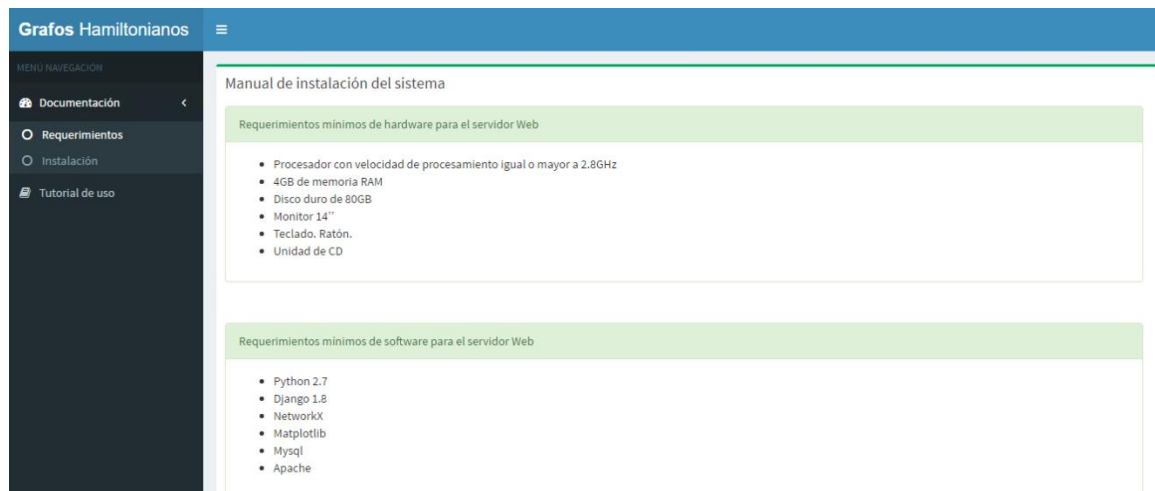


Figura C1. Vista de la sección de requisitos e instalación en la interfaz del manual de usuario Web

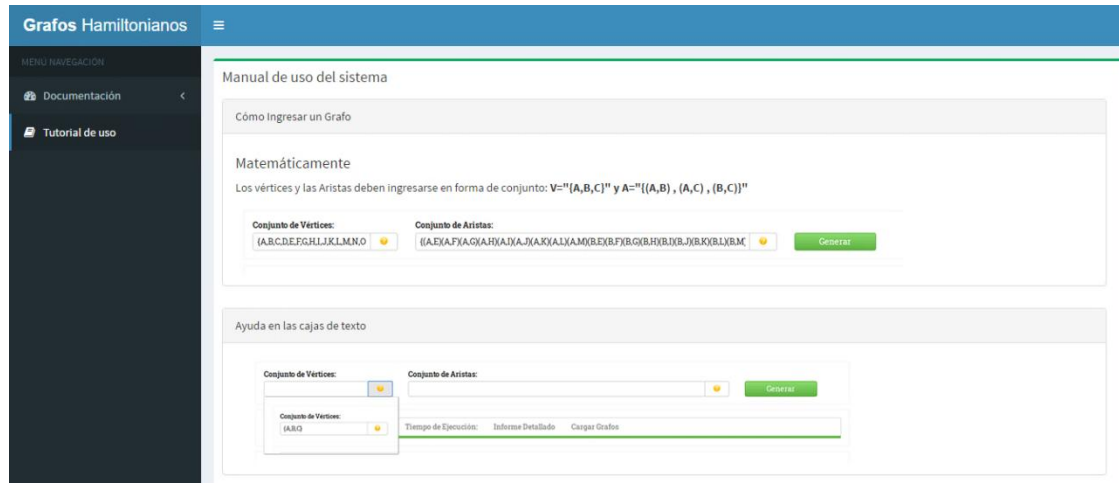


Figura C2. Vista de la sección tutorial de uso en la interfaz del manual de usuario Web

Apéndice D: Manuales de la aplicación

Manual de instalación del sistema

Requerimientos mínimos de hardware para el servidor Web

- Procesador con velocidad de procesamiento igual o mayor a 2.8GHz.
- 4GB de memoria RAM.
- Disco duro de 80GB.
- Monitor 14’’.
- Teclado. Ratón.
- Unidad de CD

Requerimientos mínimos de software para el servidor Web

- Sistema operativo Windows 7 o Debian 7.
- Manejador de base de datos Mysql.
- Apache
- Python 2.7
- Django 1.8
- NetworkX
- Matplotlib

Instalación del sistema

1. Realizar una instalación estándar de *Python 2.7*.
2. Realizar una instalación estándar de *Apache*.
3. Realizar una instalación estándar de *Mysql*.
4. Realizar una instalación estándar de *Django 1.8*.
5. Realizar una instalación estándar de *NetworkX*.
6. Realizar una instalación estándar de *Matplotlib*.
7. Copiar la carpeta hamiltonicidad del CD en una ubicación pública de su preferencia.
8. Cree la base de datos colocando el nombre: BDhamiltoniano.
9. Restaure el archivo respaldo de la base de datos “grafos” del CD.

10. Abrir el archivo de configuración dentro de la carpeta hamiltonicidad (hamiltonicidad/hamiltonicidad/settigs.py) , ir a la sección DATABASES y cambiar los siguientes valores

```
'default': {  
  
    'ENGINE': 'django.db.backends.mysql',  
    'NAME': 'BDhamiltoniano',  
    'USER': 'root',  
    'PASSWORD': 'COLOCAR SU CONTRASEÑA',  
    'HOST': 'localhost',  
    'PORT': '3386',  
}
```

11. Ir a la carpeta hamiltonicidad donde se encuentra la aplicación y ejecutar el comando por consola (python manage.py runserver) para ejecutar la aplicación.

Uso del sistema

Para la utilización y manejo adecuado de la aplicación Web desarrollada para determinar la hamiltonicidad de grafos bipartitos balanceados, se debe contar con los equipos de computación y software necesarios para su funcionamiento.

Manual de Usuario Final

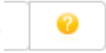
Como ingresar un grafo

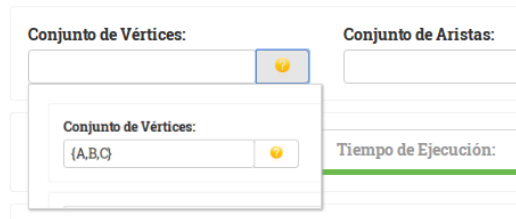
La aplicación permite el ingreso de grafos por medio de conjuntos matemáticos y gráficamente.

Ingresar grafos en forma matemática:

Para ingresar un grafo en su forma matemática debe ingresar el conjunto de vértices y el conjunto de aristas a través de las cajas de textos:

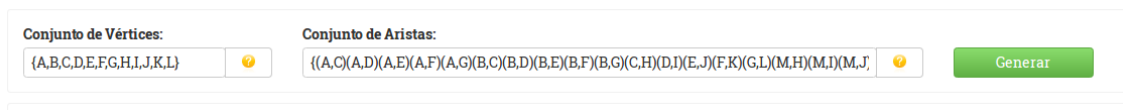
Conjunto de Vértices:	Conjunto de Aristas:	
		Generar

El icono  muestra un breve ejemplo de la forma en cómo debe ingresar los vértices y aristas.



Para ingresar los vértices debe comenzar con una “{” seguido de los vértices separados cada uno por una “,” y culminar con el cierre “}”. Ejemplo: {a,b,c,d}.

Para ingresar las aristas debe comenzar con una “{” seguido de los pares de vértices separados por una “,” y encerrados en “()” y culminar con el cierre “}”. Ejemplo: {(a,b) (c,d) (a,d)}.



Si escribe de forma errónea algunos de los campos la aplicación le mostrara un mensaje de error indicándole en cual de ambos conjuntos se ha equivocado.

Por último, darle clic al botón generar para realizar los cálculos.

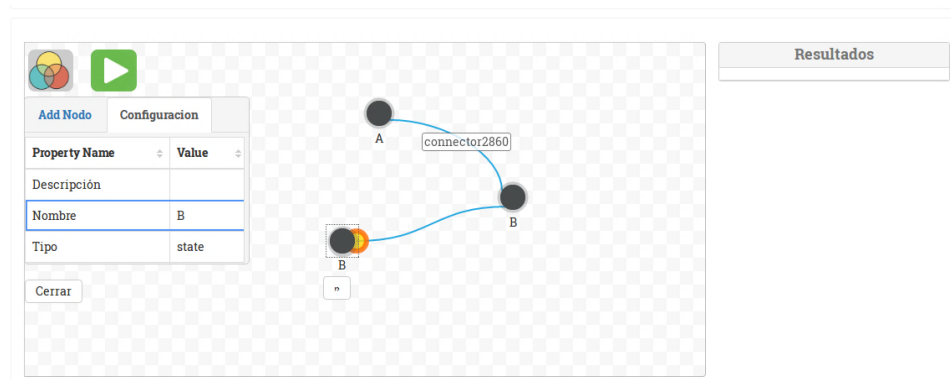
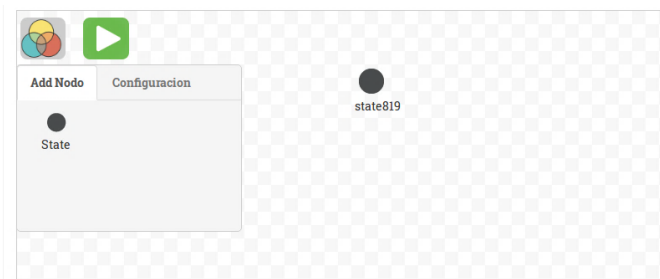
Generar

Ingresar grafos en forma gráfica:

Para ingresar grafos en forma gráfica darle clic al botón



Arrastre los nodos hacia la caja de la derecha, colocando el ratón encima de ellos conecte los nodos que desee. Edite el nombre de cada nodo y colóquele etiquetas de un caracter.



Para realizar los cálculos presione el botón.



Interpretación de resultados:

La siguiente sección muestra los resultados de los cálculos: mínimo grado, orden del grafo, tiempo de ejecución, conjuntos disjuntos. Los conjuntos disjuntos son mostrados de colores diferentes.

Conjunto de Vértices:

{A,B,C,D,E,F,G,H,I,J,K,L,M,N}

Conjunto de Aristas:

{(A,C)(A,D)(A,E)(A,F)(A,G)(B,C)(B,D)(B,E)(B,F)(B,G)(C,H)(D,I)(E,J)(F,K)(G,L)(M,H)(M,I)(M,J)}

Generar

Mínimo Grado: 3

Orden del Grafo: 14

Tiempo de Ejecución: 0.0685269832611 s

Informe Detallado

Mostrar Grafico

Cargar Grafos



Resultados

Orden del Grafo

Mínimo Grado

14

3



Grafo Hamiltoniano

Conjuntos Disjuntos

El Grafo es Bipartito Balanceado

Conjunto A

CEDGFMN

Conjunto B

ABIHKJL

VER DETALLE

Mensajes de interacción con el usuario

Si el grafo ingresado no cumple con el teorema le mostrará el siguiente mensaje, con el icono.



En caso de cumplir con el teorema mostrará el siguiente mensaje, con el siguiente icono.



Ver detalladamente los resultados:

Para ver de forma más detallada los resultados presione el botón “Ver detalle”.

En esta sección, puede ver a detalle todas las posibles combinaciones de conjuntos independientes que se pueden formar con el grafo, en el caso de no cumplir con el teorema, puede ver el conjunto que no cumple con la condición.

Conjunto de Vértices:

{A,B,C,D,E,F,G,H,I,J,K,L,M,N}

?

Conjunto de Aristas:

{{(A,C)(A,D)(A,E)(A,F)(A,G)(B,C)(B,D)(B,E)(B,F)(B,G)(C,H)(D,I)(E,J)(F,K)(G,L)(M,H)(M,I)(M,J),

?

Generar

Mínimo Grado: 3

Orden del Grafo: 14

Tiempo de Ejecución: 0.0685269832611 s

Informe Detallado

Mostrar Grafo

Cargar Grafos

Resultado detallado:

Sea G el Grafo ingresado

Conjunto de Vértices:

V={A,B,C,D,E,F,G,H,I,J,K,L,M,N}

Conjunto de Aristas:

E={{(A,C)(A,D)(A,E)(A,F)(A,G)(B,C)(B,D)(B,E)(B,F)(B,G)(C,H)(D,I)(E,J)(F,K)(G,L)(M,H)(M,I)(M,J)(M,K)(M,L)(N,H)(N,I)(N,J)(N,K)(N,L)(F,L)(E,L)(E,H)(G,J)(C,J)}

Detalle del Grafo G

Orden	Mínimo Grado	Cota del Teorema	Bipartito Balanceado
14	3	5	

Grafo Hamiltoniano

Conjuntos Disjuntos Balanceado del Grafo G

Conjunto A

CEDGFMN

Conjunto B

ABIHKJL

Conjuntos Independientes Balanceado

[u'CEIK', u'CDKL', u'CGIK', u'EGIK', u'DGHK', u'DFHJ', u'GFIH', u'MNAB]

Cargar y Guardar grafos:

Al momento de realizar los cálculos puede guardar el grafo ingresado para posteriores pruebas, dando clic en el botón “Guardar resultados”.

Resultados

Orden del Grafo

Minimo Grado

12

3



No cumple con las condiciones de la conjetura

VER DETALLE

GUARDAR RESULTADOS

Para cargar grafos de la base de datos, seleccione la opción “Cargar Grafos” de la barra de navegación

Conjunto de Vértices:
{A,B,C,D,E,F,G,H,I,J,K,L,M,N}

Conjunto de Aristas:
{(A,C)(A,D)(A,E)(A,F)(A,G)(B,C)(B,D)(B,E)(B,F)(B,G)(C,H)(D,I)(E,J)(F,K)(G,L)(M,H)(M,I)(M,J)}

Generar

Minimo Grado: 3

Orden del Grafo: 14

Tiempo de Ejecución: 0.0685269832611 s

Informe Detallado

Mostrar Grafico

Cargar Grafos

En esta sección vera el listado de grafos cargados, haga clic en el botón “Cargar” para cargarlos en la caja de texto superior, y luego presione generar para realizar los cálculos.

Conjunto de Vértices:

Conjunto de Aristas:

Generar

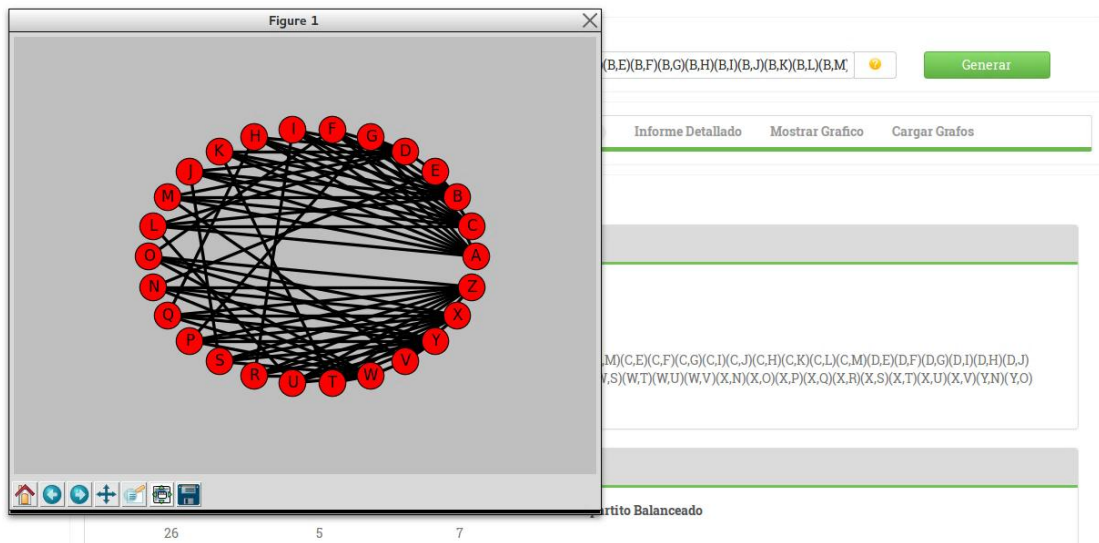
Listado de Grafos Almacenados para estudiar

Nodos:
{A,B,C,D,E,F,G,H,I,J,K,L}

Edges:
{(A,C)(A,D)(A,E)(A,F)(A,G)(B,C)(B,D)(B,E)(B,F)(B,G)(C,H)(D,I)(E,J)(F,K)(G,L)(M,H)(M,I)(M,J)(M,K)(M,L)(N,H)(N,I)(N,J)(N,K)(N,L)(F,L)}

Cargar

La opción mostrar gráfico del menú, genera el gráfico del grafo ingresado



ANEXOS



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-N°034/2009

El Consejo Universitario de la Universidad de Oriente, en uso de las atribuciones que le confiere el artículo 18 del Reglamento de la Universidad de Oriente, y con el voto unánime de todos sus miembros,

RESUELVE

ÚNICO: Aprobar el Reglamento de Trabajo de Grado de Pregrado de la Universidad de Oriente para ser aplicado a partir del II semestre de 2009, el cual quedará redactado de la siguiente forma:

REGLAMENTO DE TRABAJO DE GRADO DE PREGRADO

CAPÍTULO I

DISPOSICIONES GENERALES

Artículo 1: Para optar al Título de Pregrado en la Universidad de Oriente es requisito parcial, en las carreras que así lo establezcan en su diseño curricular, aprobar un Trabajo de Grado, en el campo correspondiente a los estudios profesionales respectivos.

Artículo 2: El Trabajo de Grado constituye un aporte teórico y/o práctico en el cual el estudiante deberá demostrar competencia para estudiar un tema y/o problema de investigación, inherente a su área profesional, a partir de la aplicación de un método en el contexto de una diversidad paradigmática.

Artículo 3: La Universidad de Oriente adopta las siguientes modalidades de Trabajo de Grado:

- a. Tesis de Grado
- b. Pasantía de Grado
- c. Cursos Especiales de Grado.

Teléfono sólo un minuto en cada oficina.

Apartado de Correos 094 / Telis: 4008042 - 4008044 / 8008045 / Telefax 4008043 / Cumaná - Venezuela



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-N° 034/2007

-2-

- Parágrafo Primero:** La Tesis de Grado constituye un aporte teórico y/o práctico, en el cual el aspirante a obtener el título profesional deberá demostrar competencia en el estudio de un tema y/o problema de investigación, mediante la aplicación de una rigurosa metodología que puede ser: descriptiva, estadística, de campo, documental, etnográfica, hermenéutica o fenomenológica.
- Parágrafo Segundo:** La Pasantía de Grado es una actividad de carácter sistemático, planificado y organizado, que permite al pasante obtener una experiencia práctica en la Universidad de Oriente, empresa o institución pública o privada, donde aplique los conocimientos teóricos y metodológicos adquiridos y desarrolle habilidades y destrezas propias de su campo profesional.
- Parágrafo Tercero:** Los Cursos Especiales de Grado se definen como un grupo de seminarios integrados en un área específica del conocimiento, destinados a profundizar la formación profesional.
- Parágrafo Cuarto:** El Consejo de Escuela o Instituto respectivo decidirá cuál (es) de esta (s) modalidad (es) se aplicará (n), de acuerdo con la planificación académica presentada por los Departamentos.
- Parágrafo Quinto:** El número de créditos del Trabajo de Grado oscilará entre 6 y 9 créditos.
- Artículo 4:** Se consideran proyectos multidisciplinarios aquellas Tesis de Grado que, dada la complejidad de los objetivos, requieran el concurso de más de un estudiante de diferentes disciplinas del conocimiento, para lograr la solución a la problemática o realidad planteada.

.../



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-N° 034/2009

-3-

Artículo 5: Para optar al Trabajo de Grado en las modalidades de Tesis de Grado, Pasantía de Grado o Cursos Especiales de Grado, avalado por el Consejo de Escuela respectivo, el estudiante deberá cumplir con los requisitos establecidos en los planes de estudio y con las exigencias internas de las unidades académicas.

Artículo 6: En las modalidades Tesis de Grado y Pasantía de Grado el estudiante deberá presentar un Proyecto ante la Comisión de Trabajo de Grado respectiva, según el instructivo y formatos establecidos al respecto.

Parágrafo Único: Se establece como condición *sine qua non* para la inscripción de la asignatura Trabajo de Grado, la aprobación del proyecto de Trabajo de Grado por la respectiva Comisión.

Artículo 7: Para la ejecución y aprobación del Trabajo de Grado en las modalidades Tesis de Grado y Pasantías de Grado se establece un (1) semestre académico como mínimo y dos (2) como máximo para su presentación ante el jurado calificador.

Parágrafo Único: Los estudiantes que no cumplan con los periodos establecidos en los planes de estudios tendrán el derecho de solicitar por escrito y ante la Comisión de Trabajo de Grado en las modalidades Tesis de Grado y Pasantía de Grado una prórroga de un (1) semestre académico avalada por el tutor para culminar el Trabajo de Grado. De no cumplir con este requisito, los estudiantes tendrán que tramitar un nuevo Trabajo de Grado sometido a la normativa vigente.

Artículo 8: Los periodos indicados para la ejecución y aprobación de cada una de las modalidades se contarán a partir de:

- a. La fecha cuando la Comisión de Trabajo de Grado apruebe el proyecto, en el caso de Tesis de Grado y Pasantía de Grado.
- b. El inicio del semestre académico regular, en el caso de Cursos Especiales de Grado.

Tráase sólo un asunto en cada oficio.



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-N° 034/2009

-4-

Parágrafo Único: La presentación y defensa del Trabajo de Grado se efectuará dentro del período regular de funcionamiento de la universidad, establecido por el calendario académico correspondiente.

Artículo 9: La calificación definitiva del Trabajo de Grado será APROBADO o REPROBADO.

CAPÍTULO II

DE LOS TRÁMITES

Artículo 10: Las solicitudes de Trabajo de Grado serán presentadas por los estudiantes, según el instructivo y formatos establecidos, ante la Comisión de Trabajo de Grado correspondiente.

Artículo 11: La entrega de las solicitudes y los trámites de aprobación se efectuarán según el instructivo que la Comisión de Trabajo de Grado suministrará a cada estudiante aspirante.

CAPÍTULO III

DE LAS COMISIONES

Artículo 12: En cada carrera funcionará una Comisión de Trabajo de Grado, integrada por un mínimo de tres (3) miembros, que será designada por el Consejo de Escuela a proposición del Director de la Escuela respectiva, oída la opinión del Jefe de Departamento de la carrera. Dicha comisión será presidida por un coordinador elegido dentro de su seno.

Parágrafo Primero: Los miembros de la Comisión de Trabajo de Grado deben ser escogidos entre el personal docente de mayor categoría escalafonaria.

...

Trátese sólo un asunto en cada oficio.



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-Nº 034 / 2009

-5-

Parágrafo Segundo: La designación como miembro de la Comisión de Trabajo de Grado será de aceptación obligatoria.

Parágrafo Tercero: Los miembros de la Comisión de Trabajo de Grado durarán un año en sus funciones, pudiendo ser ratificados, por un periodo adicional, nuevamente por el Consejo de Escuela respectivo.

Artículo 13: La Comisión de Trabajo de Grado tendrá las siguientes atribuciones:

- a) Verificar que el aspirante cumpla con los requisitos exigidos por el presente reglamento y los instructivos correspondientes, para someter a su consideración el proyecto de Trabajo de Grado.
- b) Recibir y evaluar los proyectos de Trabajo de Grado (modalidades Tesis de Grado y Pasantía de Grado), velando que los proyectos estén relacionados con líneas de investigación de la carrera y/o con los planes de desarrollo del Estado Venezolano, tanto en el ámbito nacional como regional para lo cual tendrá un lapso máximo de veinte (20) días hábiles.
- c) Notificar por escrito al tutor y al estudiante de los resultados de la evaluación del proyecto.
- d) La Comisión de Trabajo de Grado respectiva, designará a dos miembros principales y dos suplentes, quienes conjuntamente con el tutor o asesor académico integraran el jurado del Trabajo de Grado en las modalidades de Tesis de Grado o Pasantía de Grado y someterán su nombramiento a la aprobación del Consejo de Escuela respectivo.
- e) Recibir y evaluar las solicitudes referentes a Cursos Especiales de Grado y proponer, si fuese el caso, las modificaciones, que a juicio de la Comisión sean necesarias.

.../

Trátese sólo un asunto en cada oficio.



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-Nº 034/2009

-6-

- f) Evacuar las consultas concernientes a los Trabajos de Grado que se formulen ante esta instancia
- g) Evaluar y mantener actualizados folletos informativos sobre los planes y proyectos de investigación de la Institución, que sirvan de guía para las diferentes modalidades.
- h) Analizar las solicitudes de prórroga, acompañadas por sus respectivos soportes.
- i) Publicar la fecha, hora y lugar de la presentación pública del Trabajo de Grado.
- j) Elaborar el acta final de presentación de Trabajo de Grado.
- k) Velar por el cumplimiento del presente reglamento y los instructivos respectivos sobre la materia.

CAPÍTULO IV

DE LA MODALIDAD TESIS DE GRADO

Artículo 14: La Tesis de Grado puede ser individual o en equipo, cuando la índole del problema a estudiar así lo requiera y será realizada bajo la supervisión y orientación de un profesor en calidad de tutor.

Artículo 15: El tutor será propuesto por el estudiante y aprobado por la Comisión de Trabajo de Grado, quien notificará la decisión al Consejo de Escuela.

Parágrafo Primero: El tutor notificará por escrito de su aceptación ante la Comisión de Trabajo de Grado.

Parágrafo Segundo: En la Tesis de Grado, puede haber más de un tutor, en cuyo caso sólo uno de ellos será responsable directo ante la UDO.

.../



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-N° 034/2009

-7-

Parágrafo Tercero: El tutor y/o cotutores, que no pertenezcan a la Institución, deberán presentar la documentación que los acrediten como profesionales con conocimientos en el área en la cual van a efectuar la tutoría.

Artículo 16: Cuando un tutor no pueda continuar con la tutoría, deberá renunciar por escrito, y de manera razonada, ante la Comisión correspondiente, la cual informará al estudiante y al Consejo de Escuela, para que se tomen las previsiones de una nueva designación.

Parágrafo Único: En caso de que el tutor no pueda continuar con la tutoría y no le sea posible renunciar debido a su ausencia personal, el estudiante solicitará por escrito, el cambio de tutor ante la Comisión.

Artículo 17: Las funciones del tutor académico serán las siguientes:

- a) Asesorar al (a los) estudiante (s) en la elaboración del proyecto de Tesis de Grado.
- b) Apoyar al (a los) estudiante (s) en el desarrollo y cumplimiento de los objetivos del proyecto de Tesis de Grado.
- c) Hacer seguimiento a la Tesis de Grado bajo su tutoría y exigir al (a los) estudiante (s) informes periódicos sobre el avance de la ejecución del proyecto.
- d) Revisar el informe final con suficiente antelación a su presentación.
- e) Notificar por escrito a la Comisión de Trabajo de Grado que la tesis está apta para su presentación y discusión ante el jurado evaluador.

Parágrafo Único: El tutor será co-responsable de la calidad científica y académica de la Tesis de Grado.

Téñese sólo un asiento en cada oficio.



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-N° 034/2009

-8-

Artículo 18: Cuando la Tesis de Grado no revele avances satisfactorios, el tutor lo notificará por escrito a la Comisión de Trabajo de Grado con copia al tesista para dejar constancia de la situación.

Artículo 19: Si durante el desarrollo de la Tesis de Grado se considera necesario un cambio del título o de los objetivos aprobados originalmente por la Comisión, el (los) estudiante (s) deberá (n) tramitar los cambios ante esta Comisión en carta razonada y avalada por el tutor responsable, anexando copia de la aprobación del proyecto original. La Comisión procesará los cambios solicitados y emitirá la notificación permanente.

CAPÍTULO V

DE LA MODALIDAD PASANTÍA DE GRADO

Artículo 20: El Trabajo de Grado, en su modalidad Pasantía de Grado debe ser individual y será realizado bajo la supervisión y orientación de un profesor de la Universidad de Oriente, en calidad de tutor académico y otro por la empresa, institución u organismo público o privado, en calidad de tutor externo, quien debe ser profesional universitario.

Artículo 21: El tutor académico será propuesto por el estudiante y aprobado por la Comisión de Trabajo de Grado quien notificará al Consejo de Escuela la decisión. La empresa, institución u organismo público o privado designará al tutor externo.

Parágrafo Primero: El tutor académico notificará por escrito de su aceptación ante la Comisión de Trabajo de Grado.

Parágrafo Segundo: El tutor externo deberá presentar la documentación que lo acredite como profesional con conocimiento en el área en la cual va a efectuar la tutoría.

.../

Trátese sólo un asunto en cada oficina.



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-Nº 034 / 2009

-9-

Artículo 22: Cuando un tutor no pueda continuar con la tutoría, deberá renunciar por escrito, y de manera razonada, ante la Comisión correspondiente, la cual informará al estudiante y al Consejo de Escuela, para que se tomen las previsiones de una nueva designación.

Parágrafo Único: En caso de que el tutor no pueda continuar con la tutoría y no le sea posible renunciar debido a su ausencia personal, el estudiante solicitará por escrito, el cambio de tutor ante la Comisión.

Artículo 23: Las funciones de los tutores académicos y externos serán las siguientes:

- a) Asesorar al (a los) estudiante (s) en la elaboración del proyecto de Pasantía de Grado.
- b) Apoyar al (a los) estudiante (s) en el desarrollo y cumplimiento de los objetivos del proyecto de Pasantía de de Grado.
- c) Hacer seguimiento a la Pasantía de Grado bajo su tutoría y exigir al (a los) estudiante (s) informes periódicos sobre el avance de la ejecución del proyecto.
- d) Revisar el informe final con suficiente antelación a su presentación.
- e) Notificar por escrito a la Comisión de Trabajo de Grado que la pasantía está apta para su presentación y discusión ante el jurado evaluador.

Parágrafo Único: Los tutores académicos y externos serán co-responsables de la calidad científica y académica de la Pasantía de Grado.

Artículo 24: El pasante de grado deberá presentar a sus tutores un informe mensual de las actividades realizadas en el ejercicio de la Pasantía.

.../



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-N° 034 / 2009

-10-

Artículo 25: El pasante de grado que, sin causa justificada, abandone su responsabilidad, se considerará reprobado y deberá optar por un nuevo Trabajo de Grado para cumplir con este requisito.

Parágrafo Único: En caso de abandono justificado o no, éste debe ser notificado por los tutores a la Comisión respectiva.

Artículo 26: Cuando la Pasantía de Grado no revele avances satisfactorios, uno o ambos tutores podrán notificar por escrito a la Comisión de Trabajo de Grado con copia al pasante para dejar constancia de la situación.

Artículo 27: Las relaciones entre la Universidad de Oriente y las empresas que concedan Pasantías de Grado, en principio deberán estar regidas por convenios.

Parágrafo Primero: En caso de que no exista convenio alguno, la empresa deberá garantizar los medios de realización del (los) proyecto(s).

Parágrafo Segundo: La Comisión deberá tener un archivo actualizado sobre los aspectos señalados en el Parágrafo Primero. Según sea el caso, velará continuamente por su cumplimiento y propondrá los cambios que considere favorables a la institución.

CAPÍTULO VI

DEL JURADO CALIFICADOR

Artículo 28: El Trabajo de Grado en las modalidades de Tesis de Grado y Pasantía de Grado tendrá un jurado calificador, conformado por el tutor, quien actuará como coordinador, dos miembros principales y dos suplentes.

Artículo 29: Si por causa justificada, un miembro principal no puede ejercer la función de jurado, éste debe renunciar por escrito ante la Comisión

...



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-Nº 034/2009

-11-

respectiva dentro de los quince (15) días siguientes a la notificación, para que se convoque al suplente.

Artículo 30: Los miembros del jurado dispondrán de quince (15) días hábiles para el estudio del Trabajo de Grado, en la modalidad de Tesis de Grado o Pasantía de Grado, contados a partir de la fecha de entrega del mismo.

Parágrafo Primero: Cuando sea necesario convocar a uno o más jurados suplentes, éste (os) dispondrá (n) de siete (7) días hábiles para la lectura del trabajo final. Si esta fecha sobrepasara a la de la presentación y defensa, se deberá hacer una nueva convocatoria para la nueva defensa del Trabajo de Grado.

Parágrafo Segundo: Ningún jurado suplente podrá hacer las veces de principal sin la autorización de la Comisión de Trabajo de Grado respectiva.

Artículo 31: Si algún miembro del jurado, sin causa justificada, no se presenta a la exposición del Trabajo de Grado, el día y hora fijados para tal fin, el coordinador de la Comisión correspondiente lo notificará por escrito al Jefe de Departamento.

Parágrafo Único: Cuando por causas imprevistas la presentación del Trabajo de Grado no pueda realizarse, la Comisión conjuntamente con el jurado calificador establecerán una nueva fecha y lo notificarán al estudiante y al Consejo de Escuela respectivo.

Artículo 32: Las copias del informe consignadas a la Comisión de Trabajo de Grado para su discusión final, serán remitidas en los próximos tres (3) días hábiles a los miembros del jurado. A su vez, éstos dispondrán un máximo de quince (15) días hábiles para aceptar o sugerir modificaciones del Trabajo de Grado.

Artículo 33: Una vez que el jurado considere el Trabajo de Grado apto, para su discusión, el coordinador de la Comisión correspondiente publicará en

Trátese sólo un asunto en cada oficio.



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

-12-

RESOLUCIÓN CU-Nº 034/2009

cartelera y notificará por escrito a los miembros del jurado, el lugar, fecha y hora de la presentación del mismo, con una antelación mínima de tres (3) días hábiles respecto a la fecha de realización de la misma.

CAPÍTULO VII

DE LA PRESENTACIÓN DEL TRABAJO DE GRADO MODALIDAD TESIS DE GRADO O PASANTÍA DE GRADO

Artículo 34: Una vez presentado y defendido el Trabajo de Grado por el (los) estudiante (es), el jurado deliberará en privado y le asignará la calificación. El Trabajo de Grado evaluado y aprobado, que requiera mejoras, deberá ser corregido por el (los) interesados (s) y entregado nuevamente, en un lapso no mayor de veinte (20) días continuos, al jurado calificador, quien asentará su veredicto en el acta correspondiente.

Artículo 35: Cuando el Trabajo de Grado tenga méritos relevantes el jurado otorgará la mención honorífica correspondiente y recomendará su publicación, dejando constancia de ello en acta.

Artículo 36: El Trabajo de Grado que sea reprobado en la discusión podrá ser presentado por el (los) interesado (s) para una última presentación y defensa en los próximos noventa (90) días.

Parágrafo Primero: Cuando un Trabajo de Grado es reprobado por segunda vez, el estudiante deberá cambiar de tema o de modalidad y tramitar una nueva propuesta de acuerdo con lo pautado en este reglamento.

Parágrafo Segundo: Si un estudiante, sin causa justificada, no se presenta a la discusión de su Trabajo de Grado el día y la hora fijados, se considerará reprobado.

.../



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-Nº 034/2009

-13-

- Artículo 37:** Las decisiones del jurado serán inapelables y las evaluaciones individuales del mismo serán confidenciales.
- Artículo 38:** El tutor académico entregará el acta al coordinador de la Comisión de Trabajo de Grado de la escuela, una vez que sean consignadas las copias definitivas en extenso y en formatos digitales con los modelos, prototipos o maquetas si los hubiere.
- Artículo 39:** La Dirección de Escuela enviará el veredicto a Control de Estudios, donde una vez transcrita la información al Registro Académico del estudiante, se le entregará un ejemplar certificado, previa identificación, y se remitirán copias a la Comisión de Trabajo de Grado, al departamento académico y a la Secretaría General de la Universidad de Oriente.
- Artículo 40:** Las copias del Trabajo de Grado, debidamente firmadas por el jurado calificador, deben ser entregadas por el departamento respectivo de la siguiente manera: un (1) ejemplar para el Departamento, un (1) ejemplar para la Biblioteca del Núcleo y un (1) ejemplar para cada miembro del jurado.
- Artículo 41:** Los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario, para su autorización.

CAPÍTULO VIII

DE LOS CURSOS ESPECIALES DE GRADO

- Artículo 42:** Los Cursos Especiales de Grado ofrecidos en cada especialidad se agruparán por áreas de conocimiento. Cada área podrá estar conformada por un máximo de cuatro (4) seminarios y deben ser, previamente avalados por el Consejo de Escuela, y aprobadas por el Consejo de Núcleo respectivo.

Apartado de Correos 094 / Telis: 4008042 - 4008044 / 8008045 / Teletex 4008043 / Cumana - Venezuela

Trátese sólo un asunto en cada oficio.



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-Nº 034/2009

-14-

- Artículo 43:** El número total de créditos de los Cursos Especiales de Grado debe cubrir el número de créditos del Trabajo de Grado de la carrera del estudiante.
- Artículo 44:** Los Cursos Especiales de Grado a ofrecerse serán programados por la unidad académica respectiva.
- Artículo 45:** Para la selección e inscripción en los Cursos Especiales de Grado, el estudiante deberá llenar la Planilla de Preinscripción, la cual será retirada en la unidad académica correspondiente, y anexar documentación que demuestre la aprobación del 70% de carga crediticia total.
- Artículo 46:** La selección en los Cursos Especiales de Grado serán efectuadas por cada unidad académica, la cual informará a la Comisión de Trabajo de Grado de la Escuela y la inscripción se formalizará ante el Departamento de Admisión y Control de Estudio del Núcleo, en un lapso no mayor de quince (15) días después de formalizado el proceso de inscripción de alumnos regulares.
- Artículo 47:** Los Cursos Especiales de Grado podrán ser administrados de forma colegiada y se establece un máximo de veinte (20) estudiantes por sección en cada uno de ellos.
- Artículo 48:** La asistencia a los seminarios es de carácter obligatorio y está sujeta a las resoluciones vigentes de la Universidad de Oriente sobre esta materia.
- Artículo 49:** En cada uno de los Seminarios, el estudiante deberá presentar, al final de cada periodo académico, un informe escrito de investigación parcial de las actividades realizadas. Este se presentará en forma individual o por equipos de trabajo.
- Artículo 50:** La evaluación adoptará formas, medios y procedimientos de acuerdo con la naturaleza y estructura de los seminarios, en concordancia con lo establecido en el Artículo 9 de este Reglamento.

Tráese sólo un suunto en cada oficio.



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

RESOLUCIÓN CU-N° 034/2007

-15-

Artículo 51: Los Cursos Especiales de Grado no tendrán reparación y sólo pueden repetirse por una (1) vez.

Artículo 52: El estudiante que resultare reprobado por dos (2) veces en un Curso Especial de Grado deberá cambiarse de área de conocimiento u optar por otra alternativa de grado.

Parágrafo Único: El cambio de área de conocimiento será permitido una sola vez.

Artículo 53: La Comisión de Trabajo de Grado expedirá una constancia de aprobación de los Cursos Especiales de Grado, cuando el estudiante haya cursado y aprobado los cursos correspondientes al área de conocimiento, y la copia será remitida al Departamento de Admisión y Control de Estudios del Núcleo.

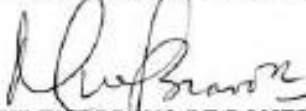
CAPÍTULO IX

DISPOSICIONES FINALES

Artículo 54: Se derogan todas aquellas disposiciones que en esta materia colidan con el presente reglamento.

Artículo 55: Lo no previsto en el presente reglamento y las dudas que surjan respecto a la interpretación y aplicación de sus disposiciones serán resueltas por el Consejo Universitario.

Dado, firmado y sellado en la sala de reuniones de Postgrado del Núcleo de Nueva Esparta, a los once (11) días del mes de junio de dos mil nueve (2009).


DRA. MILENA BRAVO DE ROMERO
Rectora-Presidenta




JUAN A. BOLANOS CURVELO
Secretario

JABC/YGG/mnuja

Apartado de Correos 094 / Telfs: 4008042 - 4008044 / 8008045 / Teletex 4008043 / Cumaná - Venezuela

Trátese sólo un asunto en cada oficio.

HOJA DE METADATOS

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 1/6

Título	Aplicación para determinar la hamiltonicidad de grafos bipartitos balanceados
Subtítulo	

Autor(es)

Apellidos y Nombres	Código CVLAC / e-mail	
Alzolar M., María de los Á.	CVLAC	19.980.225
	e-mail	alzolarma@gmail.com
	e-mail	alzolarma@yahoo.es

Palabras o frases claves:

Aplicación
Hamiltonicidad
Grafos Bipartitos Balanceados
Teoría de Grafos
Conjunto independiente de vértices

Líneas y sublíneas de investigación:

Área	Subárea
CIENCIAS	Informática

Resumen (abstract):

Se desarrolló una aplicación para determinar la hamiltonicidad de grafos bipartitos balanceados en función de su mínimo grado y número de vecinos, ofreciendo funcionalidades para manipular grafos de características específicas, entre ellas; grafos simples y de orden $2n$. Para esta investigación se tomó como referencia la metodología CA/IO (Ciencia de la Administración e Investigación de Operaciones) la cual consta de 6 fases: identificación, observación y planteamiento del problema; donde se obtuvo los requisitos de la aplicación estudiando el modelo matemático planteado. Construcción del modelo, en donde se plantearon las soluciones del problema en cuestión, y se planteó un árbol de decisión con los pasos a seguir. Generación de una solución, en el cual se construyó los algoritmos de cada proceso de la aplicación guiándose por el árbol de decisión planteado. Prueba y evaluación de una solución, en el que se evaluaron los procesos y se dio fe de su veracidad. Implante y evaluación. Se implementó PERT/CPM para la planificación de la misma. La aplicación obtenida ofrece a investigadores y a estudiantes del área de estudio una herramienta para facilitar el estudio de los grafos bipartitos y mejorar el desempeño de los mismos al agilizar el proceso.

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 3/6

Contribuidores:

Apellidos y Nombres	ROL / Código CVLAC / e-mail	
Carmen Romero	ROL	CA ☐ AS ☒ TU ☐ JU ☐
	CVLAC	10.947.403
	e-mail	cvromero@gmail.com
Daniel Brito	ROL	CA ☒ AS ☐ TU ☐ JU ☐
	CVLAC	3.823.342
	e-mail	danieljosb@gmail.com
Claudia Carmona	ROL	CA ☐ AS ☐ TU ☐ JU ☒
	CVLAC	13.275.156
	e-mail	claudiacarmona0610@gmail.com
Felicía Villarroel	ROL	CA ☐ AS ☐ TU ☐ JU ☒
	CVLAC	10.203.708
	e-mail	feliciavillarroel@gmail.com

Fecha de discusión y aprobación:

Año	Mes	Día
2016	05	27

Lenguaje:

Español

Archivo(s):

Nombre de archivo	Tipo MIME
Tesis_Maria_Alzolar.docx	Aplication/Word

Alcance:

Espacial: Universal (Opcional)

Temporal: Intemporal (Opcional)

Título o Grado asociado con el trabajo:

Licenciado en Informática

Nivel Asociado con el Trabajo:

Licenciado

Área de Estudio:

Sistemas de Información (SI)

Institución(es) que garantiza(n) el Título de grado:

Universidad de Oriente (UDO)



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

CUN°0975

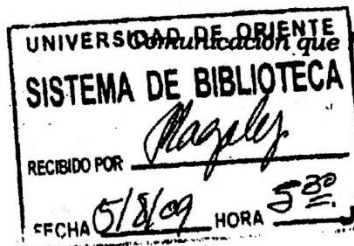
Cumaná, 04 AGO 2009

Ciudadano
Prof. JESÚS MARTÍNEZ YÉPEZ
Vicerrector Académico
Universidad de Oriente
Su Despacho

Estimado Profesor Martínez:

Cumplo en notificarle que el Consejo Universitario, en Reunión Ordinaria celebrada en Centro de Convenciones de Cantaura, los días 28 y 29 de julio de 2009, conoció el punto de agenda **"SOLICITUD DE AUTORIZACIÓN PARA PUBLICAR TODA LA PRODUCCIÓN INTELECTUAL DE LA UNIVERSIDAD DE ORIENTE EN EL REPOSITORIO INSTITUCIONAL DE LA UDO, SEGÚN VRAC N° 696/2009"**.

Leído el oficio SIBI – 139/2009 de fecha 09-07-2009, suscrita por el Dr. Abul K. Bashirullah, Director de Bibliotecas, este Cuerpo Colegiado decidió, por unanimidad, autorizar la publicación de toda la producción intelectual de la Universidad de Oriente en el Repositorio en cuestión.



Comunicación que hago a usted a los fines consiguientes.

Cordialmente,

JUAN A. BOLANOS CUNEL
Secretario



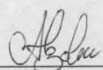
C.C: Rectora, Vicerrectora Administrativa, Decanos de los Núcleos, Coordinador General de Administración, Director de Personal, Dirección de Finanzas, Dirección de Presupuesto, Contraloría Interna, Consultoría Jurídica, Director de Bibliotecas, Dirección de Publicaciones, Dirección de Computación, Coordinación de Teleinformática, Coordinación General de Postgrado.

JABC/YGC/maruja

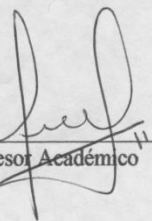
Apartado Correos 094 / Telf: 4008042 - 4008044 / 8008045 Telefax: 4008043 / Cumaná - Venezuela

Derechos:

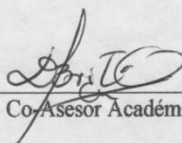
Artículo 41 del REGLAMENTO DE TRABAJO DE PREGRADO (vigente a partir del II Semestre 2009, según comunicación CU-034-2009): “los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario para su autorización”.



Autor



Asesor Académico



Co-Asesor Académico