

14
2ej.

01168

**ANALISIS POR TEORIA DE GRAFICOS
DEL
MODELO EDUCATIVO BASADO EN
COMPETENCIAS.**

**Tesis para obtener el grado de
Maestro en Ingenieria
(Investigación de Operaciones)**

Tomás Oliver Sánchez

UNAM - DEPI 1998

264556

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas

Tesis Digitales

Restricciones de uso

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTO.

Quiero expresar mi agradecimiento al Dr. Ricardo Aceves García por el ánimo y ayuda profesional que me prestó para realizar este trabajo. Doy igualmente gracias al M. en I. Arturo Fuentes Zenón por el interés que se tomó en el desarrollo de este trabajo. Mi reconocimiento a la Lic. Beatriz Jasso Arizmendi por su eminencia en el área pedagógica únicamente superada por su generosidad, ella proporcionó la información referente al Modelo Educativo Basado en Competencias y dedicó largo tiempo dándome asesorías y comentarios sobre el tema.

Por último, pero no en menor medida aprecio la ayuda, los comentarios y sugerencias de los profesores de la Escuela Superior de Ingeniería y Arquitectura y del C.E.T. No. 1 "Ing. Walter Cross Buchanan".

INDICE.

■ CAPITULO 1.	
Antecedentes del MEBC.....	1
■ CAPITULO 2.	
El Modelo Educativo Basado en Competencias (MEBC).....	15
■ CAPITULO 3.	
Aplicación del MEBC (Centro Experimental de Educación Basada en Competencias en IPN.).....	33
■ CAPITULO 4.	
La Teoría de Gráficas (Una introducción).....	47
■ CAPITULO 5.	
Análisis del MEBC (CET No. 1 Walter Cross Buchanan) aplicando la Teoría de Gráficas.....	117
■ CAPITULO 6.	
Conclusiones.....	147
■ APENDICE A.	
Algoritmos de gráficas programados en Lenguaje C.....	161
■ APENDICE B.	
Ejemplo.....	227
■ BIBLIOGRAFIA.....	247

COMPENDIO

Al establecer un conjunto de actividades a realizar normalmente se ejecutan con determinada secuencia y se busca que ésta sea óptima bajo ciertos criterios pero cuando se establecen relaciones de actividades que generan aprendizaje o conocimientos la secuencia para realizarlos, se considera generalmente, que queda definida por la lógica o el pensamiento vertical en el cual esta basado el desarrollo científico y técnico. Cuando las actividades a desarrollar son de carácter académico se generan vinculaciones internas entre ellas las cuales nos producen efectos que no por ostensibles pero que si impactan de alguna manera en mayor o menor grado al beneficiario del producto de aquellas actividades y por lo tanto estos efectos no deben de ser soslayados. Por la naturaleza combinatoria de las relaciones de actividades es susceptible dar un tratamiento con la Teoría de Gráficos, la vinculación interna entre actividades se le conoce como síntesis de conocimiento y han sido asociadas a los circuitos en gráficas [15]

En el I.P.N. se estableció como plan piloto el Modelo Educativo Basado en Competencias (MEBC) en la escuela C.E.T. N°1 Walter Cross Buchanan, esta implantación consistió en la realización de actividades docentes y administrativas enmarcadas en cinco características generales.

- 1) EJES ESTRUCTURANTES.
- 2) CRITERIOS METODOLOGICOS PARA EL DISEÑO DE PLAN DE ESTUDIOS.
- 3) AREAS FUNCIONALES DEL CENTRO EXPERIMENTAL.
- 4) SERVICIOS EDUCATIVOS.
- 5) ALCANCES DEL MODELO.

Con estas actividades podemos definir los objetivos de este proyecto.

Objetivo 1.

Establecer una relación entre las características elementos de la característica 1 (Ejes estructurantes) y los elementos de la característica 2 (Alcances del modelo).

Objetivo 2.

Determinar la combinación de efectos, síntesis de conocimientos, según la relación de los elementos de las actividades 1 y 2.

Para lograr el objetivo número dos es importante reforzar ciertas relaciones que resultaran relevantes lo cual podemos plantearlo como un objetivo más.

Objetivo 3.

Determinación de las relaciones a reforzar que definen las Areas de Síntesis de conocimiento y análisis.

Estas Areas de Síntesis de conocimiento relacionándolas (circuitos básicos), a su vez, generan otras áreas de síntesis las cuales también es interesante conocerlas y analizarlas, con lo cual establecimos el siguiente objetivo.

Objetivo 4.

Determinar áreas de conocimiento inducidas.

Cumpliendo con estos cuatro objetivos se termina el análisis del MEBC. Es obvio que con estos objetivos planteados no se tienen agotadas todas las potencialidades del MEBC, conscientes de esta situación y como una invitación al lector potencial de este trabajo a continuar en la búsqueda de atributos del MEBC, se presentan algunos planteamientos como cortes y cortes básicos así como una serie de algoritmos programados en lenguaje C de la Teoría de Gráficos.

CAPÍTULO 1

1. INTRODUCCIÓN

A nivel internacional continúa la tendencia hacia la globalización de la economía, la búsqueda de mayor competitividad en los mercados de bienes y servicios, incluyendo los financieros, y el acelerado cambio tecnológico que no solo comprende la tecnología **dura** (hardware), o sea la base técnica de la producción, sino también, a lo que se le conoce como tecnología **blanda** (software), es decir, las formas en que se organiza y administra la producción y el trabajo.

La globalización no solo implica la necesidad de mantener la estabilidad de las principales variables económicas, como finanzas públicas, tipo de cambio, inflación y política comercial, sino también es necesario enfrentar el reto que significa la globalización de la fuerza de trabajo.

La movilidad del trabajo ha exigido mayor velocidad de cambio y adaptación de los sistemas de educación y capacitación a las tendencias tecnológicas y a las transformaciones organizativas de la producción.

Por ello, la globalización incluye además de los procesos de producción y comercio, la de los sistemas de educación y capacitación de recursos humanos por lo que estos deben mejorar su calidad y acelerar su adaptación a las nuevas condiciones de la tecnología, la economía y sociedad.

En este contexto, la capacitación no puede concebirse como una actividad aislada y desvinculada de la estrategia y del mejoramiento de la productividad y competitividad de la economía, ni tampoco de la de ampliación de los niveles de bienestar de la población.

Por el contrario la capacitación constituye un proceso esencial para la transformación productiva y el desarrollo con equidad social.

Las Secretarías de Educación Pública y del Trabajo y Previsión Social en septiembre de 1993 iniciaron conjuntamente un proyecto sobre Educación Tecnológica y Modernización de la capacitación del cual surgió la necesidad de establecer los Sistemas de Certificación de Competencia laboral en México, desde sus inicios el proyecto contó con la participación y apoyo de los sectores obrero y empresarial.

Trabajo y Previsión Social acordaron modificar los programas de capacitación Técnica y para ello se fijó el siguiente objetivo:

Mejorar la calidad de la educación Técnica y la capacitación de manera pertinente y flexible, lo cual se logrará con el Sistema Normalizado de Competencias Laborales (SNCL) y de su certificación. El complemento del SNCL es la Educación Basada en Normas de Competencias.

El Colegio Nacional de Educación Profesional Técnica (CONALEP) junto con las Direcciones Generales de Educación Tecnológica Industrial y de Centros de Formación para el Trabajo, fueron seleccionados por las Secretarías de Educación Pública y del Trabajo y Previsión Social para llevar a cabo pruebas piloto con las que se daría inicio la adaptación de esta nueva modalidad educativa en México. El CONALEP fue elegido por dos razones fundamentales: el desarrollo de los programas de vinculación con el sector productivo y la flexibilización de su oferta educativa que se llevo a cabo desde 1992, fecha en que se instauró el programa modular, el cual se caracteriza por una curricula desarrollada a partir de contenidos tecnológicos en función de puestos de trabajo específicos y de mayor flexibilidad.

El CONALEP llevó a cabo diversos estudios de las experiencias de otros países en materia de programas de capacitación basados en el establecimiento del Sistema de Normalización de Competencia Laboral (SNCL) y para ello estudió con especial interés el sistema inglés pionero en este tema. Para el complemento del SNCL, la Educación Basada en Normas de Competencia se analizaron los modelos que están funcionando en la provincia de Quebec en el Canadá, en el estado de Oklahoma de los Estados Unidos de Norteamérica y recientemente se han tomado en cuenta los de Australia, Francia y Alemania y en especial la modalidad de educación en alternancia, ampliamente difundida en estos dos últimos países.

Dichos estudios han enriquecido la planeación de las experiencias que el CONALEP ha efectuado y aportan elementos considerables en el diseño de un modelo propio y adecuado a la cultura productiva de México.

Las aportaciones vertidas por los distintos estudios y convenios efectuados con organismos internacionales permitieron conocer y precisar los cambios que habrían de desarrollarse en la implementación de la Educación Basada en Normas de Competencias, tanto estructurales como organizacionales, pedagógicas y financieras.

La cooperación internacional entre el CONALEP y organizaciones gubernamentales y privadas, le ha permitido contar con el apoyo de instituciones que están a la vanguardia de los avances educativos y tecnológicos en esta nueva modalidad. Para la EBNC destacan los siguientes convenios y aportaciones.

■ Consejo Británico de Inglaterra (1992).

Aportaciones:

- 1) Desarrollo de metodologías para el establecimiento de normas de competencias que fueran reconocidas a nivel internacional, así como los criterios de certificación que una persona debe tener para ser capaz de efectuar una o varias tareas en un trabajo determinado de acuerdo con normas establecidas por el sector productivo.
- 2) Programas de estudio en las áreas de Ingeniería de Mantenimiento, Manufactura Industrial y Desarrollo Profesional.
- 3) La instrumentación de programas conjuntos de formación de instructores.

■ Universidad Estatal de Oklahoma (1994).

Aportaciones:

- 1) Adquisición de 20 materiales curriculares de las áreas de electrónica, computación, mantenimiento de computadoras personales y soldadura.

■ Canadá (1994).

Aportaciones :

- 1) Transferencia de las metodologías utilizadas en este país para la definición de competencias laborales; como la del Análisis Funcional del Trabajo; y los criterios para el diseño de los planes de formación (carreras).
- 2) Formación de instructores para la modalidad de EBNC.

Suministro de 81 guías de aprendizaje en las áreas de Mecánica Automotriz, Asistente Ejecutivo y Electromecánica de Sistemas Automatizados.

- 3) Evaluación del desempeño del personal docente.

Gracias a estos convenios y sobre las bases anteriores se puso en operación la primera prueba piloto de esta modalidad educativa.

Es importante hacer mención del proceso de planeación, desarrollo y administración de los programas de formación profesional en Quebec, debido a la aportación del Canadá por medio de los convenios con México se tiene mucha influencia en la implantación del nuevo modelo educativo en las instituciones educativas mexicanas.

En general, el programa de desarrollo y programas de formación en Quebec incluye cuatro fases:

- a) Planeación.
- b) Desarrollo.
- c) Puesta de marcha.
- d) Evaluación.

Estas cuatro fases se inscriben en un proceso continuo, es decir, que un programa cualquiera se sitúa siempre dentro de un proceso. Teniendo en cuenta la rápida evolución de las situaciones de trabajo (Cambios tecnológicos, cambios en el mercado laboral, etc.) los programas deberían ser evaluados cada cinco o seis años.

Por otra parte, como la participación de las empresas es uno de los principios básicos del enfoque por competencias, en cada fase participan comités de consulta o grupos de trabajo.

a) Fase de Planeación.

La fase de planeación de los programas de formación se inicia después de un proceso de planeación estratégica. Es el momento de decidir si se desarrolla o se revisa un programa de formación. Dado que los programas se agrupan por sector de formación, en primer lugar, la planeación de los programas de formación debe inscribirse en el marco de una planeación por sector de formación. Para cada uno de éstos se procede a estudios que permitan determinar los programas por desarrollar o revisar. Precisemos que el estudio de un sector de formación no es idéntico al estudio de un sector económico, que es responsabilidad de una autoridad diferente y tiene una finalidad diferente.

b) Fase de desarrollo.

Durante esta fase se redactan tanto el programa de formación como un documento (provisional) en que se desarrolla el programa.

También durante esta fase se conciben y elaboran los documentos que servirán de guía para quienes participarán en la puesta en marcha del programa: formadores, directores de centros de formación, evaluadores, desarrolladores de material didáctico, alumnos, etcétera.

El desarrollo de un programa consiste en determinar los comportamientos esperados, los criterios de evaluación y las condiciones en que éste se llevará a cabo.

c) Fase de puesta en marcha.

Durante esta fase se preparan los documentos oficiales y el programa es sometido a la realidad de la enseñanza y el aprendizaje.

d) Fase de evaluación.

Cualquier situación de evaluación se inscribe en una perspectiva de toma de decisiones. No obstante se puntualiza que desde la perspectiva de los programas, la evaluación incluye diferentes situaciones en el ámbito de los objetivos de evaluación, en el de decisiones tomadas y en el de los medios utilizados para recabar la información necesaria para tomar la decisión. Esto significa que el objetivo de la evaluación no puede separarse de la naturaleza misma de la decisión que se va a tomar, como tampoco la formación recabada ni los medios utilizados con este fin.

Así pues, ya sea que se desee tomar la decisión sobre la enseñanza o sobre los aprendizajes, o incluso sobre los programas mismos, las etapas de estas fases de evaluación serán diferentes y se precisa que "tanto la evaluación del aprendizaje como la de la enseñanza forman parte de la evaluación del programa".

El Instituto Politécnico Nacional tiene como propósito tener implantado, en toda la Institución el Modelo Educativo Basado en Competencias (MEBC) ^[1] en cinco años más y tomó como escuela piloto al Centro Educativo No. 1 Walter Cross Buchanan, por haber estado funcionando

^[1] Originalmente en el CONALEP se adoptó el término EBNC pero al implantar el nuevo modelo educativo en el IPN, se cambió el término por MEBC.

con un esquema similar al MEBC. Las características principales de este esquema son:

- a) Ha ofertado carreras para personas mayores de 18 años.
- b) Flexibilidad en la duración de sus estudios.
- c) Pertinencia en las necesidades del sector productivo.
- d) Reconocimiento con previa evaluación de las competencias adquiridas en el sector laboral.
- e) Salidas laterales.
- f) Revalidación de estudios.

Las fases programáticas que se consideran para la implantación del MEBC son diez:

Fase 0

Compilación, estudio de información y programación de actividades.

Procedimiento:

Técnicas de investigación documental y de campo. Métodos inductivos, deductivos, analógico, hipotético y analítico-sintético.

Producto:

Métodos, procedimientos, técnicas, estrategias, programas de trabajo y guías de operación.

Fase 1

Integración e inducción de equipos de trabajo.

Procedimiento:

Información documental, conferencias, seminarios y asesorías.

Producto:

Células de trabajo, programas de trabajo, control de avance, control de calidad.

Fase 2

Identificación de competencias.

Procedimiento:

Análisis funcional y de situaciones reales de vida, de estudio y de trabajo.

Producto:

Mapas funcionales, elementos y unidades de competencia.

Fase 3

Determinación de normas de competencia.

Procedimiento:

Reglas generales y específicas del “conocer”.

Producto:

Normas técnicas de competencia y módulos de aprendizaje.

Fase 4

Elaboración de programas de estudio.

Procedimiento:

Análisis estructural de tareas y secuencia epistémica conforme a cuatro tipos de conocimiento.

Producto:

Programas de estudios y métodos e instrumentos de evaluación.

Fase 5

Instrumentación didáctica y evaluativa.

Procedimiento:

Aplicación del paradigma constructivista, del aprendizaje, normas técnicas de competencia y guías para la evaluación.

Producto:

Actividades de aprendizaje, de evaluación y de instrucción.

Fase 6

Integración de materiales educativos.

Procedimiento:

Aplicación de la tecnología cognoscitiva de textos y de imágenes y de las modernas tecnologías de comunicación e información.

Producto:

Textos, láminas, videos, modelos, programas de cómputo, paquetes didácticos y otros recursos técnicos de aprendizaje e información.

Fase 7

Formación y evaluación de docentes y preparación de estudiantes.

Procedimiento:

Talleres didácticos, cursos, seminarios y conferencias.
Impresos, videos, evidencias de conocimiento y desempeño.

Producto:

Profesores técnicamente preparados para facilitar el aprendizaje de normas de competencia. Estudiantes responsables de su aprendizaje dispuestos a evaluarse conforme a estándares de calidad.

Fase 8

Aplicación piloto y validación de programas.

Procedimiento:

Desempeño monitoreado de estudiantes y profesores integrados en módulos de aprendizaje conforme a programas de estudio de normas de competencia.

Producto:

Detección y corrección de desviaciones, ajustes a procedimientos y obtención de datos para la planeación estratégica y la toma de decisiones.

Fase 9

Diseño y operación del sistema de información IPN-EBNC.

Procedimiento:

Integración de centro de información documental especificación de conceptos, procesos y productos informáticos; creación y explotación de la base de datos.

Producto:

Datos organizados sobre: estudiantes y profesores; normas de competencia, programas de estudio, módulos de aprendizaje, resultados de evaluación y materiales educativos.

Considerando la filosofía de la Educación Basada en Normas de Competencias y las características y necesidades económicas sociales y de producción se diseñó el Modelo Educativo Basado en Competencias (MEBC) para el Centro Educativo Tecnológico No. 1 del IPN el cual se presenta en el capítulo 3. En el capítulo 2 se tiene una propuesta general para la implantación del MEBC en México.

Como observaremos en los capítulos 3 y 4 el MEBC para el C.E.T. No 1 podemos establecer relaciones entre los diferentes componentes del modelo, en varios niveles, pero se pretende realizar en éste trabajo un análisis de las relaciones en la estructura del modelo, por medio de la Teoría de Gráficas, en lo referente a las áreas de síntesis, relaciones inducidas, circuitos inducidos y dualidad.

Este análisis como se da a conocer en el capítulo 5 resulta conveniente puesto que puede determinar relaciones y efectos según la topología que se establece en una gráfica que represente los elementos estructurales del MEBC.

De ninguna manera se pretende en éste trabajo, agotar todas las posibilidades de análisis en el MEBC, el afán primordial es dar una propuesta técnica por medio de la Teoría de Gráficas que ayude a la implantación del MEBC en el IPN en particular en la Escuela Superior de Ingeniería y Arquitectura (ESIA).

En el capítulo 4 se da una introducción de los conceptos y resultados básicos de la Teoría de Gráficas, suficientes para poder desarrollar este trabajo en lo referente al análisis.

El capítulo 6 contiene un resumen de los resultados obtenidos así como una serie de conclusiones y comentarios.

Finalmente tenemos dos apéndices, en el A un conjunto de algoritmos programados en lenguaje C, éstos algoritmos básicamente determinan rutas, árboles de expansión, circuitos de Euler y de Hamilton, Matriz de Circuitos Fundamentales y Matriz de Cortes Básicos necesarios para el análisis propuesto. Los algoritmos fueron tomados de la bibliografía especializada en Teoría de Gráficas y en algunos casos se hicieron modificaciones respecto a la lógica propuesta, la programación es responsabilidad del autor de éste trabajo. Y en el ápendice B tenemos un ejemplo resumido del análisis realizado en el capítulo 6.

CAPÍTULO 2

2. EL MODELO EDUCATIVO BASADO EN COMPETENCIAS.

Introducción.

Existe consenso internacional en que los incrementos en la habilidad, talento y conocimiento de las personas en edad productiva, han demostrado ser eficaces como contribuyentes mayores en el crecimiento económico; y la alteración en la distribución del capital humano, como talento práctico, parece ser el método político preferido para eliminar tanto las barreras de pobreza como las de ingreso de la población.

No es accidental que la modalidad educativa basada en competencias este sugerida casi al mismo tiempo en países con diferentes grados de desarrollo y con gobiernos de diferentes inclinaciones políticas, tales como Gran Bretaña, Estados Unidos, Canadá, Australia y Nueva Zelanda.

El cambio tecnológico en los medios de producción, buscando la competitividad entre los países, ha alterado necesariamente las condiciones técnicas de la fuerza de trabajo y en consecuencia, la necesidad de elevar los niveles de educación para la población económicamente activa.

Existe, ahora más que nunca, incertidumbre acerca del papel de la educación en el mundo post-moderno, por lo que el movimiento de **Educación Basada en Competencias** deberá plantearse en México como una política educativa que conjugue la teoría con la práctica y la educación técnica con la educación académica general, con el decidido afán de transformar a los educandos, no tan solo en productores y consumidores, sino en seres humanos comprometidos con el cambio de estructuras sociales, políticas y económicas, de una manera consciente del papel que les corresponde jugar en la actualidad.

La **Educación Basada en Competencias** se nos presenta como una alternativa viable, que: mediante la definición de competencias

fundamentales para la formación personal, intelectual y axiológica de nuestros educandos; de las competencias transferibles que permitan la libre distribución del capital humano en nuestros sectores productivos; y de las competencias laborales que hagan posible el desarrollo de habilidades, conocimientos destrezas y actitudes económicamente útiles, intenta lograr la superación educativa que reclama la población de nuestro país.

En la actualidad existen observaciones del sector productivo hacia el sector educativo respecto a las diferencias que hay entre los conocimientos, habilidades, destrezas y actitudes que muestran los egresados de la educación técnica y los requerimientos específicos de las empresas para el desempeño de su trabajo, lo que ocasiona que éstas tengan necesidad de establecer sus programas propios de reentrenamiento que satisfagan dichos requerimientos, con el consiguiente desperdicio de recursos por rectificación de la capacitación.

Por lo anterior, la reestructuración de los servicios de educación tecnológica y capacitación técnica, deberá realizarse tomando en cuenta los requerimientos del sector productivo, sobre la base de estándares de competencias definidas por el propio aparato productivo y certificadas por una instancia pública como lo serán los sistemas normalizados de Competencia y Certificación Laboral, que se describirán más adelante.

También se deberá estructurar un sistema de normas y acciones que aseguren a los trabajadores una pertinente respuesta a sus necesidades y posibilidades educativas, por lo que la capacitación se ofrecerá a través de módulos flexibles en duración y horarios, que les proporcionen competencias que faciliten su incorporación al mercado laboral y los habiliten para desarrollarse en él, pero que además le sirvan de base para continuar estudiando una carrera técnica de mayor duración.

México ha venido generando una profunda reforma en el terreno económico, incrementando su eficiencia en sus relaciones económicas internas y con otros países, como lo es el Tratado de Libre Comercio con Estados Unidos y Canadá lo cual nos obliga a buscar la conformación de una fuerza laboral apta y competente para alcanzar una mayor productividad, calidad y competitividad, como parte de la estrategia de desarrollo del país; lo que a su vez, se espera reflejar en mejores niveles de vida para la población.

Una fuerza laboral tal, deberá ser el objetivo a lograr a través de la reforma de los servicios de educación tecnológica y capacitación técnica, establecidos para sustentar el desarrollo de nuestro país, incrementando la calidad educativa mediante un sistema de Educación Basado en Competencias Laborales.

El proyecto de reforma de los servicios de educación tecnológica y capacitación técnica en México considerará cuatro sistemas intercalando coordinadamente a nivel nacional, como un Sistema Normalizado de Competencia Laboral definido en torno a la identificación de estándares y normas de competencia, con diferentes niveles de complejidad y rasgos de transferibilidad. El otro sistema componente será el Sistema Nacional de Certificación de Competencia Laboral conformado como un sistema de evaluación que se aplicará en todas las áreas de competencia laboral, que calificará diversos niveles de desempeño, cubriendo los requerimientos mínimos de validez y confiabilidad de resultados, así como la equidad e imparcialidad en su aplicación. El tercer sistema componente que interactuará en el proyecto de reforma, es el encargado de formar a los usuarios del Sistema Nacional de Certificación, que serán aquellos egresados de los sistemas de educación tecnológica y de los centros de capacitación del sector productivo, así como a los prácticos laborales formados empíricamente a través de su experiencia en el trabajo.

Los resultados del Sistema Nacional de Certificación de Competencia Laboral serán los técnicos capacitados reconocidos, que el sector productivo podrá emplear confiablemente estableciendo así criterios que permitan la contratación, retención, promoción, reubicación capacitación o actualización de su personal.

El proyecto de reforma mencionado, tendrá como eje principal las necesidades que tiene el sector productivo de personal técnicamente calificado, identificado éste en términos de sus competencias laborales, que se describirán de manera conductual, como: las habilidades, conocimientos, actitudes y destrezas necesarios para desempeñar un trabajo específico del mercado laboral, a niveles predeterminados y normalizados de ejecución, establecidos con la participación directa del sector productivo.

Las diferencias entre Educación Basada en Competencias y la educación tradicional, se describen en la tabla 2.1 en que se comparan las características entre ambas.

Los resultados obtenidos en la aplicación de la **Educación Basada en Competencias** en otros países, reportan beneficios significativos en la formación de recursos humanos, como son:

- Los empleadores tienen la oportunidad de participar directamente en la determinación del entrenamiento requerido, para satisfacer sus necesidades específicas de recursos humanos.
- Los empleadores participan también en la determinación de las habilidades que tienen mayor probabilidad de ser adquiridas en el lugar de trabajo.
- Los empleadores pueden conocer con anticipación el nivel de competencias que van a lograr los alumnos a través de los planes de estudio.
- Los estudiantes tienen un sentido de dirección más completo, sabiendo lo que deben aprender y su forma de evaluación.
- Los estudiantes tienen la seguridad de que lo aprendido es aplicable a un futuro empleo en el campo de trabajo de su elección.
- Los estudiantes aprenden a medida que aseguran en sus evaluaciones, calificaciones más altas.
- Los estudiantes aplican por tiempo más largo lo aprendido.
- Mayor número de estudiantes alcanzan altos grados de destreza.
- Los estudiantes experimentan el éxito de su aprendizaje desde el principio del programa, obteniendo mayor motivación y aceptación del sistema.
- Los estudiantes aprenden a responsabilizarse de su propia capacitación.

DIFERENCIAS ENTRE EDUCACIÓN BASADA EN COMPETENCIAS Y EDUCACIÓN TRADICIONAL

<u>PROCESOS</u>	<u>EDUCACIÓN TRADICIONAL</u>	<u>EDUCACIÓN BASADA EN COMPETENCIAS</u>
Admisión de alumnos.	Una vez al año.	En cualquier momento durante el año.
Reinscripciones.	Una o dos veces al año. Fechas fijas de inicio para todos los estudiantes.	Cada vez que un alumno inicia un nuevo módulo.
Horarios.	Horarios rígidos mantenidos durante el semestre. Horarios establecidos por la escuela.	Horarios flexibles basados en la disponibilidad de los profesores, de las instalaciones y del equipo.
Contenido del Programa.	Enfocado principalmente a la adquisición de conocimientos académicos (científicos y tecnológicos).	Dirigidos al dominio de competencias formativas y requeridas por el mercado de trabajo.
Metodología de Enseñanza.	Centrada en el profesor. En su mayoría, expositivo a grandes grupos de estudiantes. Los estudiantes toman nota de lo escrito en el pizarrón por el profesor. La calidad del aprendizaje depende mayormente de la calidad y disposición del profesor. En grandes grupos, solamente algunos de los estudiantes demuestran sus habilidades prácticas.	Enfocada al estudiante. El profesor sirve como un medio para clarificar dudas, demostración de habilidades y proporcionar materiales de aprendizaje. Materiales instruccionales de alta calidad proveen las bases para el aprendizaje de los alumnos. Los estudiantes reciben asistencia del profesor en forma individual o en grupos pequeños cuando ésta es requerida.

Tabla 2.1. Comparación de la Educación Tradicional y la Educación Basada en Competencias.

DIFERENCIAS ENTRE EDUCACIÓN BASADA EN COMPETENCIAS Y EDUCACIÓN TRADICIONAL

<u>PROCESOS</u>	<u>EDUCACIÓN TRADICIONAL</u>	<u>EDUCACIÓN BASADA EN COMPETENCIAS</u>
Metodología del Aprendizaje.	Pasiva, atendiendo a las exposiciones en el aula o a las demostraciones en el laboratorio o también, por parte del profesor. Estudio asistemático, dependiendo de las tareas extraescolares asignadas o de los períodos de exámenes.	Activa, responsabilizándose de la consulta de información. Comprometida al logro de avances mínimos preestablecidos con el profesor. Desarrollo individualizado de habilidades mínimas, cuyos criterios están establecidos en el programa.
Evaluación.	Normativa, de acuerdo a los promedios obtenidos por el grupo. El alumno demuestra al profesor sus conocimientos normalmente por medio de pruebas objetivas escritas o de ensayo. Sin considerar el resultado, el profesor continúa con los siguientes tópicos del programa.	Objetiva. Basada en criterios (estándares). Se pone mayor énfasis en la demostración de las competencias adquiridas, apoyada con exámenes escritos. Los estudiantes deben probar ser competentes en cada módulo, antes de proceder con el siguiente.
Certificación	Expedida hasta el final del programa educativo completo.	Expedida al completar todos los requerimientos especificados para cada una de las competencias establecidas en el programa educativo.

Tabla 2.1. : Continuación.

- Se puede aprender más en el mismo tiempo. Los paquetes de materiales didácticos eliminan tiempos de espera para recibir instrucción en tareas que los estudiantes ya dominan.

- El aprendizaje a su propio ritmo, permite a los estudiantes optimizar su tiempo y determinar la ayuda que requieren de sus profesores o asesores.
- Los índices de deserción disminuyen, ya que los estudiantes experimentan su aprendizaje desde el principio, sin la presión de competir para la calificación con otros estudiantes y sin tener que esperar por semanas o meses en clases de teoría.
- Los instructores disponen de un proceso enseñanza-aprendizaje estructurado que les permite aplicar, analizar, sugerir y desarrollar recursos de aprendizaje, técnicas de enseñanza, procedimientos de prueba y requerimientos de equipamiento y de espacios educativos preestablecidos, pero factibles de mejorar por él mismo.
- Los docentes, al dejar de ser expositores, disponen de más tiempo para ayudar a los estudiantes que más lo necesitan.
- Por sus características de autodidactismo, el programa avanza por sí mismo cuando el instructor se ausenta.
- Los estudiantes tienen una participación más activa en el proceso y se mantienen ocupados durante más tiempo, buscando desarrollar sus habilidades y profundizando sus conocimientos.

El programa crea una atmósfera profesional similar al ambiente de trabajo, lo que contribuye a una mayor satisfacción para los estudiantes, instructores y administradores. La mayoría de los programas para la educación tecnológica y la capacitación técnica que han venido aplicándose en los países en vías de desarrollo, han tomado como modelo a los diseñados en países industrializados. Dichos programas fueron con frecuencia desarrollados en tiempos en que los cambios en la introducción de nuevas tecnologías, en los procesos de producción, herramientas, equipo y materiales, así como en los perfiles ocupacionales, ocurrían con mucha mayor lentitud que los cambios actuales en el mundo. La metodología y los materiales instruccionales usados para tales programas requieren de un alto nivel de competencia técnica y pedagógica por parte de los profesores, mientras que las estructuras rígidas basadas en tiempos (anuales, semestrales, etc.), tanto de los programas como de los materiales instruccionales, hacen difícil sus modificaciones para que respondan a las necesidades actuales de formación de recursos humanos para el trabajo.

Desde los primeros años de la década de los 70's, algunos países iniciaron investigaciones para desarrollar sistemas de capacitación técnica de alta flexibilidad, orientados a la formación para el empleo y apoyados por materiales **instruccionales** centrados en el alumno, que vinculados con la realidad del mercado de trabajo, dieron origen a los sistemas modulares.

La modularización de los programas de formación implicó identificar a las tareas como una división de trabajo, aceptablemente lógica de un trabajo particular u ocupación.

Existen en la actualidad tres enfoques más difundidos sobre el desarrollo del currículum basado en competencias:

- Un currículum centrado en un sistema de evaluación, en el que los mismos **estándares de competencia** han llegado a convertirse en el currículum. Estos estándares fueron inicialmente establecidos en 1986 por el Consejo Nacional para Calificaciones Vocacionales (NCVQ) en Inglaterra, País de Gales e Irlanda del Norte, pretendiendo mejorar la educación técnica en un modelo de desarrollo curricular, tratando de que las instituciones educativas desarrollaran el suyo propio, a su conveniencia.
- Otro enfoque más sistemático es el DACUM (Designing a curriculum), desarrollado a finales de los años 60's en Columbia Británica, Canadá, basado en una conceptualización de competencias orientadas hacia las tareas que describen específicamente las actividades integradas en puestos de trabajo y ocupaciones.
- Recientemente, ya en esta década, se ha desarrollado en Australia el currículum basado en problemas, el cual pretende ser un modelo que integre a los dos anteriores.

La Educación Basada en Competencias pretende ampliar el perfil profesional de tal manera que caracterice las carreras en calificaciones y las competencias generales que cada una deberá contener.

De esta manera, el perfil de las competencias, llamado en ocasiones **"récord" de habilidades o perfil de empleo**, conforma un documento que relaciona las áreas de competencia requeridas en una ocupación y provee de un medio para evaluar el dominio de competencias específicas; para tal fin, se considerarán los estándares establecidos por el Sistema Normalizado de Competencia Laboral (SNCL).

Como ya se ha dicho anteriormente, en la **Educación Basada en Competencias** el tiempo deja de ser un factor de medida para avanzar en el programa educativo, éste se sustituye por el resultado o logro de competencias conforme al ritmo individual de aprendizaje de cada participante.

Por tal motivo, el plan de estudios sólo ofrece sugerencias de **"duraciones promedio"** para el aprendizaje, las que servirán sólo como referencia para controles de avance, contratación de instructores y registro oficial de las carreras.

Para la **Educación Basada en Competencias**, los objetivos se definen como competencias descritas en términos de conocimientos, habilidades y destrezas necesarios para lograr niveles de ejecución claramente especificados y acordados por el sector productivo. Las actividades de aprendizaje y la asignación de prácticas se enfatizan más en el desarrollo de habilidades sin la presión del tiempo.

Los medios propuestos son aquellos que permiten el auto-aprendizaje, y los procedimientos de evaluación ponderan las listas de verificación que comprueban actividades y normas de eficiencia, sobre las pruebas objetivas que comprueban conocimientos.

Además, se debe proveer una variedad suficiente de experiencias de aprendizaje, tales como juegos de roles y otras actividades de simulación para ayudar al participante a dominar un conjunto mínimo de competencias, haciendo al instructor corresponsable de los logros de sus alumnos.

La operación de un Sistema de Certificación de Competencia, va aparejada con la consideración de la metodología que deberá emplearse para subsanar las deficiencias de una persona que, pretendiendo obtener un documento acredite su dominio de competencias para una ocupación o una carrera técnica.

La integración de grupos de participantes con las mismas características académicas y sicomotrices será imposible de lograr y no podrán optimizarse los tiempos de costo de instructores, tiempo, aula, laboratorio, etc. Por tanto, una institución educativa que adopte la **Educación Basada en Competencias**, de enseñanza y aprendizaje basada en el autodidactismo, que permita que los participantes satisfagan sus necesidades de conocimientos, habilidades y destrezas, requeridos por el sector productivo (competencias) , en forma individual y a su propio ritmo de aprendizaje. La individualización de la capacitación permite que el participante y no el instructor, sea el elemento central que se debe enfocar para maximizar el aprendizaje.

Los componentes básicos de una capacitación individualizada, son los contenidos de la capacitación, las estrategias, los medios auxiliares educativos y los ambientes instruccionales, de los cuales se identifican: los contenidos, los planes y programas de diseño curriculum y los medios instruccionales como aspectos tecnológicos que se emplean para mejorar el aprendizaje; las estrategias instruccionales como los procedimientos alternativos que emplea un instructor según las características de los participantes; y los ambientes instruccionales como las condiciones del aula, talleres y laboratorios que propicie el aprendizaje.

Los materiales didácticos requeridos para la individualización de la capacitación para usarse en la **Educación Basada en Competencias**, son parte de la tecnología por emplear.

A la tecnología instruccional la conforman los materiales y medios que apoyan a los participantes para el logro de las competencias requeridas.

Estos pueden ser materiales escritos con las características necesarias para el auto-aprendizaje, tales como las hojas de dirección, las guías de aprendizaje y los paquetes didácticos; los medios audiovisuales tales como acetatos, diapositivas y videos que permitan la individualización del aprendizaje supliendo a la exposición clásica del profesor. Los programas de cómputo educativos en sus funciones de simulación, podrá suplir a las demostraciones prácticas de talleres y laboratorios, observando y experimentando el funcionamiento de equipos reales y los efectos y causas de fallas de sus componentes.

El papel que desempeña el docente es fundamental para cualquier sistema educativo, puesto que de él depende en gran medida, el éxito o el fracaso de un programa. Su influencia en el grupo de alumnos es decisiva, pues aún existiendo las condiciones óptimas para la instrucción, si el docente no desempeña su labor adecuadamente, los recursos no estarán bien aprovechados. En cambio, en la situación inversa, si el instructor es eficiente y posee características como la experiencia docente y el dominio en las actividades tecnológicas, el aprendizaje estará garantizado.

El docente también deberá ser flexible para modificar las técnicas y los métodos en caso necesario y dominar el tema para el buen uso de los materiales didácticos, las instalaciones y el equipo disponibles.

La formación que debe poseer el docente dentro de la **Educación Basada en Competencias**, comprende su experiencia pedagógica en la organización de la metodología de aprendizaje; es decir, debe dominar las estrategias adecuadas para llevar a cabo el proceso de formación; poseer conocimientos sobre el comportamiento humano con la finalidad de entender que cada persona es única y especial, y que, por tanto, requiere una atención particular; realizar esquemas que permitan determinar lógicamente y por grado de complejidad los contenidos temáticos de una materia; contar con iniciativa para establecer creativamente el escenario en que se desarrolla el curso, sesión, clase o asesoría; lo que permitirá una mejor comunicación y motivación de los alumnos. Debe además, tener experiencia en el manejo y conducción de grupos, pues en cada uno de ellos, por estar conformado de personas distintas, se creará un ambiente de trabajo único e irrepetible, por lo que el docente debe poseer la capacidad para manejarlo, de tal forma, que los objetivos de la sesión sean alcanzados.

Para el logro del perfil docente debe diseñarse un programa adecuado, con el fin de proporcionarle los elementos pedagógicos, didácticos y psicológicos fundamentales, que le permitan ejercer su tarea educativa e incidir en el proceso de enseñanza-aprendizaje, mediante el dominio de metodología, técnicas y procedimientos que propicien la mejora de la calidad de la educación técnica, así como para dotarlo de las herramientas que le posibiliten una participación más activa en la evaluación, adecuación y actualización de programas de estudio y en la elaboración de textos y de material didáctico.

La formación de los docentes para la **Educación Basada en Competencias** implica un cambio de actitudes: de su función tradicional de expositor, al de orientador y guía de las actividades de los alumnos, evaluador de competencias adquiridas y controlador y suministrador de materiales didácticos y equipamiento a utilizar para los estudiantes para desarrollar sus actividades de auto-aprendizaje.

La certificación de los docentes-instructores, deberá considerar la actitud de éstos hacia el cambio de la función que tradicionalmente desempeñaban en el salón de clases, el taller o el laboratorio.

Además deberá considerarse el grado de dominio que tengan sobre las competencias definidas en el programa por impartir; aplicándose para tal fin los mismos métodos e instrumentos de evaluación que los empleados para certificar a los estudiantes. Una vez establecido el Sistema Nacional de Certificación y Competencia Laboral, quedará a cargo de este sistema el proceso de certificación de los profesores de todas las instituciones educativas de capacitación para el trabajo.

La formación de los alumnos de esta modalidad educativa, es el fin último que se persigue, como elemento fundamental en que se desarrollará la fuerza laboral apta y competente, que permita una mayor productividad, calidad y competitividad requerida para el desarrollo de nuestro país.

Su caracterización se hace como sigue:

POBLACIÓN A ATENDER:

Los grupos específicos de población a atender son:

- a) Aquellos jóvenes que buscan incorporarse al mercado de trabajo.
- b) Las personas que conforman a la población desempleada y que requieren incrementar sus posibilidades de incorporarse al mercado laboral.
- c) Aquellos trabajadores que requieren actualizar sus conocimientos, habilidades y destrezas, de acuerdo a los cambios tecnológicos introducidos en las líneas de producción.
- d) Las personas que habiendo cursado una carrera técnica, deseen ampliar o profundizar en algún campo específico del conocimiento e incrementar sus evaluaciones será determinada por cada institución, acorde con los contenidos específicos de cada competencia.

Los procesos de evaluación y certificación son los componentes básicos para identificar y reconocer oficialmente tanto las competencias de entrada con que se presentan los aspirantes como las competencias adquiridas durante el proceso de enseñanza-aprendizaje en las instituciones educativas.

Además de la información documentada que presentan los aspirantes al ingresar al sistema de Educación Basada en Competencias, la evaluación preliminar es un mecanismo necesario para identificar las competencias de entrada que un trabajador dice tener, con el fin de tener un medio para ubicarlo en el nivel de estudios adecuado a sus conocimientos, habilidades y destrezas obtenidas previamente.

Esta evaluación es la que se lleva a cabo durante el proceso enseñanza-aprendizaje, por parte del docente que imparte o coordina los contenidos y experiencias de aprendizaje de los alumnos, que permitirá retroalimentar al estudiante sobre sus avances y posibilidades del logro de la competencia.

El número y la naturaleza de las evaluaciones será determinado por cada institución, acorde con los contenidos específicos.

Esta evaluación deberá aplicarse al final del proceso de enseñanza-aprendizaje de manera centralizada por la institución educativa efectuado por un verificador externo al plantel ajeno a los docentes que impartieron o coordinaron el proceso.

El reconocimiento de las competencias demostradas por un estudiante a través de la evaluación se lleva a cabo mediante dos tipos de certificación:

- La certificación está representada por un documento oficial expedido por la institución educativa que llevo a cabo el proceso de formación del estudiante, mediante el cual se reconoce el logro de los estándares de cada competencia, después de la evaluación sumaria.
- La certificación que se llevará a cabo por los centros de certificación autorizados por el Sistema Nacional de Certificación de Competencia Laboral, después de que la persona solicitante (egresado de una institución educativa o un trabajador formado empíricamente) logra satisfactoriamente los estándares de la competencia en cuestión.

El grado en que el sector productivo se involucre en el proceso de reforma de la educación tecnológica y la capacitación técnica en nuestro país marcará sin duda, el éxito o fracaso de dicho proceso, ya que la participación de las empresas es fundamental para que la reforma opere.

El sector productivo participará en acciones conjuntas en el Sector Educativo y la Secretaría del Trabajo en:

- a) La determinación de las necesidades de capacitación y formación.
- b) La definición de las competencias y su forma de logro, que deben adquirir los trabajadores.
- c) La integración de las comisiones mixtas que conformen a los organismos de certificación.
- d) Apoyo logístico y de infraestructura para la operación del proceso educativo.

El sector productivo participará en la determinación, diseño y validación de las competencias que integrarán los programas educativos mediante la selección y envío de sus mejores trabajadores en el campo del trabajo a desarrollar.

El sector productivo participará, a través de las cámaras y la integración de los organismos de certificación, los cuales acreditarán centros de evaluación para la aplicación de instrumentos que conduzcan a la certificación, registrando sus resultados y emitiendo certificados.

Dichos organismos podrían, en su momento acreditar a centros de capacitación públicos y privados que estuvieran en condiciones de ofrecer formación adecuada que conduzcan a la certificación; podrían también establecer y desarrollar bancos de materiales didácticos para la instrucción y bancos de pruebas para la evaluación que podrían ser usados por los centros de capacitación acreditados, así también, dichos organismos podrían emplear y entrenar verificadores que vigilen la adecuada operatividad de los centros de capacitación acreditados por ellos.

CAPÍTULO 3

3. IMPLANTACIÓN DEL MEBC EN EL CET. No.1 DEL IPN.

Introducción.

El Centro Educativo Tecnológico No.1 fué seleccionado como escuela piloto en el Instituto Politécnico Nacional por haber funcionado hasta septiembre de 1996 con un model educativo similar al MEBC. Sus principales características fueron las siguientes:

- 1) Ha aportado carreras para personas mayores de 18 años.
- 2) Flexibilidad en la duración de los estudios.
- 3) Pertinencia ante las necesidades del Sector Productivo.
- 4) Con previa evaluación, reconocimiento de estudios y competencias adquiridas en el campo laboral.
- 5) Posibilidad de salidas laterales.
- 6) Revalidación de estudios de otras instituciones.

A continuación se da el esquema general del MEBC implantado en el C.E.T. No. 1, como plan piloto autorizado por las autoridades educativas del I.P.N..

Iniciamos con algunas definiciones básicas.

- a) **Pertinente** a las necesidades planteadas por el sector productivo para el desarrollo económico del país, así como por el sector educativo del nivel superior.
- b) **Flexible** en la duración de estudios de la carrera, lo cual dependerá de la disposición de tiempo y el aprovechamiento académico del alumno, dando la posibilidad a la población económicamente activa de mejorar sus competencias en horarios accesibles para ellos.
- c) **Polivalente** por la gama de opciones aceptadas para capacitarse o formarse, tales como: cursos de capacitación, salidas laterales.

CARACTERÍSTICAS GENERALES DEL MODELO.

EJES ESTRUCTURALES.

- Trinomio Educación-Sector Productivo-Sociedad.
- Currículum.
- Docencia.
- Cultura Organizacional Académica.

CRITERIOS METODOLOGICOS PARA EL DISEÑO DEL PLAN DE ESTUDIOS.

- Flexibilidad .
- Actualidad.
- Equilibrio.
- Congruencia.
- Secuencia.
- Relevancia.

ÁREAS FUNCIONALES DEL CENTRO EXPERIMENTAL.

- Centro de capacitación.
- Centro de formación propédeutica y terminal.
- Centro de educación continua.
- Centro de apoyo y servicios tecnológicos.

SERVICIOS EDUCATIVOS.

- Capacitación.
- Formación de técnicos profesionales, opción terminal.
- Formación de técnicos como opción bivalente.

- Educación continua.

ALCANCES DEL MODELO.

- Posibilidad de titulación al concluir el 6o semestre.
- Certificación de estudios por el I.P.N..
- Perfil del egresado acorde a los estándares académicos requeridos por el Sistema Nacional de Certificación de Competencias Laborales.
- Salidas laterales.
- Vinculación con la industria.
- Posibilidad de cursar una carrera técnica con bachillerato con una duración flexible.

EJES ESTRUCTURANTES

TRINOMIO EDUCACIÓN-SECTOR PRODUCTIVO-SOCIEDAD.

Este eje estructurante se conceptualiza como una "alianza estratégica" entre el sector productivo, educativo y las unidades de su contexto, basada en una serie de paradigmas que les permitan conciliar con éxito los objetivos de cada uno de ellos, encaminados a formar profesionales e investigadores en los diversos campos de la ciencia y la tecnología de acuerdo con los requerimientos del desarrollo económico, político y social de la nación y participando en la consolidación de la independencia económica, científica, tecnológica, cultural y política a través de una educación socialmente útil y nacionalmente necesaria.

CURRICULUM.

Este eje estructurante contempla la dosificación equilibrada, actual, flexible y factible de los aprendizajes, facilitando una intervención activa del alumno en su proceso formativo. Considera los criterios de flexibilidad, equilibrio, actualidad, congruencia, secuencia y relevancia en mapas curriculares.

DOCENCIA.

Este eje estructurante reconceptualiza el quehacer de los docentes desde los ejes constitutivos de su trabajo, entre estos destacan las relaciones que marcan y posibilitan su labor académica, los modelos educativos que orientan su práctica y axiología social y educativa que los identifican con los fines y valores de una institución.

CULTURA ORGANIZACIONAL ACADÉMICA.

Este eje estructurante contempla el sistema de valores, normas, conocimientos y formas de expresión que rigen el comportamiento de los miembros de nuestra comunidad académica.

CRITERIOS METODOLOGICOS ADAPTADOS PARA EL DISEÑO DEL PLAN DE ESTUDIOS.

FLEXIBILIDAD

- El plan de estudios está estructurado bajo este criterio, por módulos de formación, (sistema modular).
- La estructura modular permite para el estudiante salidas laterales, cursos de capacitación y de educación continúa.
- El plan de estudios prevé la certificación del saber demostrado.
- Adecuación de rutas y aprendizaje de módulos, de acuerdo con las posibilidades reales de tiempo del alumno.
- Posibilidad de transitar de las salidas laterales terminales a la salida - vertical, propedeútica.

EQUILIBRIO.

- 55% Contenido Tecnológico.
- 45% Contenido científico básico y humanístico.
- Integración entre las diferentes áreas del conocimiento (científicas, humanísticas y tecnológicas).
- Integración entre las diferentes competencias (tecnológicas, científicas y humanísticas), requeridas por el perfil del egresado.

ACTUALIDAD.

- Posibilidad de actualizar sistemáticamente las competencias laborales, en vinculación con el sector productivo de bienes y servicios.
- Posibilidad de incorporación de los avances científicos y tecnológicos más recientes de los contenidos del plan de estudios.

CONGRUENCIA.

- Competencias tecnológicas requeridas por el mercado de trabajo.
- Competencias científicas, básicas y humanísticas requeridas para la formación del nivel medio superior, como propedeútico del nivel superior.
- Competencias transferibles para emplearse en diferentes áreas laborales.

SECUENCIA.

- Se determinan cadenas de competencias de la misma calificación en períodos sucesivos, integrando módulos.

- Se establecen relaciones de antecedente y consecuente de las competencias y módulos.

RELEVANCIA.

- Selección adecuada de competencias para el logro de cada calificación que integra las carreras.

ÁREAS FUNCIONALES COMO CENTRO EXPERIMENTAL DE EDUCACIÓN BASADA EN COMPETENCIAS.

COMO CENTRO DE CAPACITACIÓN ofrecería:

- Competencias tecnológicas específicas.
- Cursos de capacitación para la incorporación inmediata al sector laboral.

COMO CENTRO DE INFORMACIÓN ofrecería:

- Formación de técnicos profesionales.
- Competencias Básicas de Formación Propedéutica requeridas como prerequisites para estudios de nivel licenciatura.
- Formación de técnicos.

COMO CENTRO DE EDUCACIÓN CONTINUA ofrecería:

- Competencias tecnológicas avanzadas que mantengan a los egresados actualizados en habilidades y conocimientos tecnológicos.
- Servicio permanente de capacitación y actualización de recursos humanos al sector productivo industrial y de servicios.

COMO CENTRO DE APOYO Y SERVICIOS TECNOLÓGICOS atendería:

- El desarrollo de proyectos de investigación tecnológica en forma interdisciplinaria.
- La asesoría y la elaboración de proyectos de desarrollo tecnológico en forma interdisciplinaria.
- El mantenimiento preventivo y correctivo a instalaciones y equipo de las empresas.
- El diseño y fabricación de prototipos para autoequipamiento.

MODELO CURRICULAR BASADO EN COMPETENCIAS

COMPONENTES	TIPOS DE COMPETENCIAS	CONDICIONES	ELEMENTOS
DISCIPLINARIO.	COMPETENCIAS GENÉRICAS.	Se traducen en manejo de: -Terminología. -Convenciones. -Tendencias. -Teoría. -Metodologías. -Estructuras.	-Recopilar, analizar y organizar información. -Comunicar ideas e información. -Planear y organizar actividades. -Trabajar con otros y en equipo. -Utilizar ideas y técnicas matemáticas. -Resolver problemas. -Usar la tecnología.
PROFESIONAL.	COMPETENCIAS PARTICULARES.	Vinculadas con: - Un campo de actividades específico (familias profesionales).	* Códigos y normas. * Áreas de desarrollo ocupacional. * Estructuras de comunicación. * Necesidades de desarrollo técnico-productivo. * Prospectivas del campo profesional.
PRÁCTICO - PRODUCTIVO	COMPETENCIAS PARTICULARES.	Refieren al uso de: - Materiales. - Equipos. - Herramientas.	Condiciones de trabajo concretas para definir desempeños productivos en función de cantidades y calidades.

Tabla 3.1

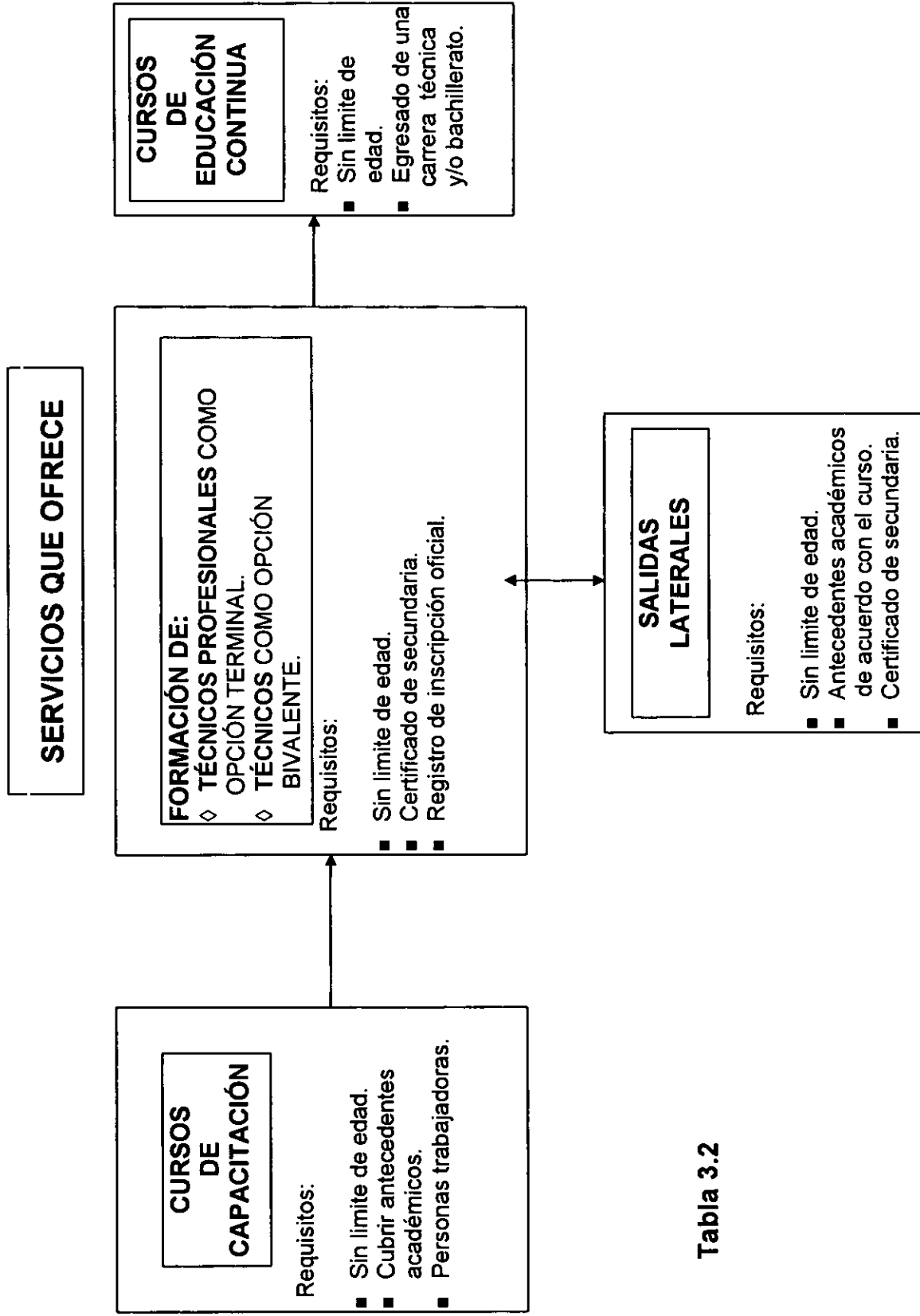


Tabla 3.2

ÁREAS DE COMPETENCIA A CONSIDERAR

Las que forman al estudiante para el trabajo (Saber Hacer):

COMPETENCIAS

⇒ LABORABLES.

TECNOLÓGICAS

⇒ TRANSFERIBLES.

Tabla 3.3

Las que forman un estudiante en un proceso educativo:

COMPETENCIAS
HUMANÍSTICAS
DE FORMACIÓN
PROPEDEUTICA

⇒ EDUCACIÓN PARA LA SALUD.

⇒ LIDERAZGO.

⇒ CREATIVIDAD E INNOVACIÓN TECNOLÓGICA.

⇒ APRECIACIÓN ARTÍSTICA.

⇒ CALIDAD DE VIDA.

Las que forman al estudiante para estudios posteriores (Saber) :

COMPETENCIAS
CIENTÍFICAS
BÁSICAS DE
FORMACIÓN
PROPEDEUTICA

⇒ MATEMÁTICAS.

⇒ CIENCIAS NATURALES.

⇒ LENGUAJES Y COMUNICACIÓN.

⇒ HISTÓRICO SOCIALES.

LÍNEAS DE ORIENTACIÓN CURRICULAR

Estas permean el currículum y están dirigidas al desarrollo de competencias (conocimientos, habilidades y actitudes) en el alumno, de tal forma que están consideradas en todos los módulos de calificación, ya sea como parte de su contenido, como área de aplicación o como parte de su instrumentación didáctica.

LÍNEAS DE ORIENTACIÓN CURRICULAR
--

- Derechos humanos.
- Habilidades del pensamiento.
- Habilidades de comunicación.
- Ecología.
- Ciencia y tecnología.
- Computación.
- Calidad.
- Seguridad e higiene.
- Economía.
- Toma de decisiones.
- Solución de problemas.
- Trabajo en equipo.

Tabla 3.4

Las líneas dan carácter de transferibilidad a las competencias clave definidas.

ESTRUCTURA CURRICULAR BASADA EN COMPETENCIAS

CARRERA TÉCNICA

COMPETENCIAS CIENTÍFICAS BÁSICAS Formación Propedéutica			
ÁREA: MATEMÁTICA	ÁREA: LENGUAJE Y COMUNICACIÓN	ÁREA: CIENCIAS NATURALES.	ÁREA: HISTÓRICO SOCIAL.



COMPETENCIAS HUMANÍSTICAS (Formación Propedéutica)			
EDUCACIÓN PARA LA SALUD	LIDERAZGO	CREATIVIDAD E INNOVACIÓN TECNOLÓGICA	APRECIACIÓN ARTÍSTICA
			CALIDAD DE VIDA



COMPETENCIAS TECNOLÓGICAS (Formación para el trabajo)			
CLASIFICACIÓN A:	CLASIFICACIÓN B:	CLASIFICACIÓN C:	
COMPETENCIA LABORAL A1	COMPETENCIA LABORAL B1	COMPETENCIA LABORAL C1	
COMPETENCIA LABORAL A2	COMPETENCIA LABORAL B2	COMPETENCIA LABORAL C2	
COMPETENCIA LABORAL An	COMPETENCIA LABORAL Bn	COMPETENCIA LABORAL Cn	



CURSOS DE CAPACITACIÓN



CURSOS DE CAPACITACIÓN



CURSOS DE CAPACITACIÓN



EDUCACIÓN CONTINUA

COMPETENCIAS TECNOLÓGICAS AVANZADAS

Tabla 3.5

CAPÍTULO 4

4. TEORÍA DE GRÁFICAS

CONCEPTOS BÁSICOS.

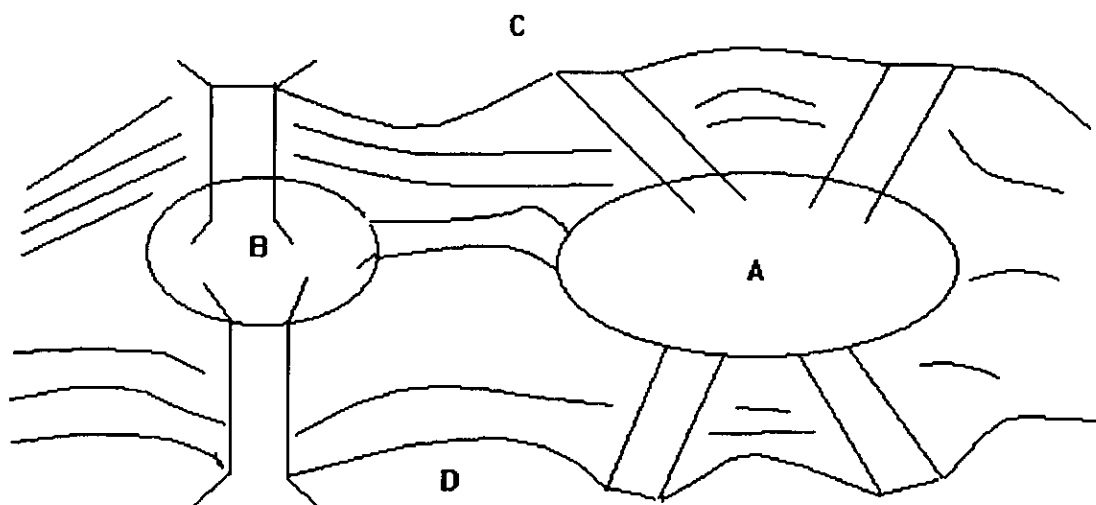
HISTORIA.

En la última década ha habido un sustancial interés en la teoría de gráficas. El resurgimiento del interés aparentemente proviene de la realización que esto ha implicado a problemas originados en diversos campos como la psicología, ingeniería, sociología, economía, antropología, lingüística y geografía. Aunque los matemáticos son responsables de mucho de este desarrollo y crecimiento, los sociólogos e ingenieros igualmente ven con entusiasmo hacia la teoría de gráficas para resolver problemas de sus campos.

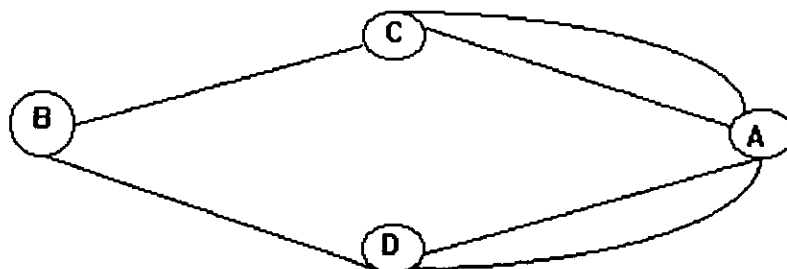
La teoría de gráficas es un fascinante sujeto. Sus orígenes son tan diversos como sus aplicaciones. El matemático suizo Leonard Euler (1707-1783) es claramente el padre de la teoría de gráficas. En la última parte del siglo XVIII, él establece un importante resultado para un famoso problema no resuelto en sus días se refiere a los 7 puentes de Königsberg, los cuales conectan dos islas y el margen de la ribera Pregel como muestra la Figura 4.1.

El problema consiste en determinar una ruta cruzando los 7 puentes una sola vez y regresar al punto de partida Euler, comprobó que esto es imposible y este hecho lo fundamento con la teoría de gráficas.

El siguiente desarrollo más importante es atribuido al físico Kirchhoff, quien en 1847 mientras estudiaba el flujo de corriente eléctrica en redes desarrolló la teoría de árboles.



(a)



(b)

**Figura 4.1 (a) : Problema del puente de Königsberg; 4.1 (b)
Representación, por medio de una gráfica.**

Cayley en su estudio con química orgánica redescubre los **árboles** cerca de 1857, estuvo implicado en la enumeración de los isómeros de hidrocarburos saturados. El problema se reduce a encontrar el número de árboles que tienen ciertas propiedades. Intentó encontrar una solución a la **conjetura de los cuatro colores** y contribuyó significativamente al desarrollo global de la teoría de gráficas; el problema establece que cuatro colores son suficientes

para colorear el mapa de un país, de tal manera que dos estados con frontera común no tengan el mismo color. Aunque este problema es fácil visualizarlo es muy difícil de resolver. En 1879 Kempe hizo una demostración. Diez años después Heawood da un ejemplo en contra para la demostración de Kempe. Considerando lo anterior, ha sido comprobado que cuatro colores son necesarios y cinco son suficientes.

En 1859, Sir William Hamilton contribuyó al crecimiento en la teoría de gráficas inventando un juego. Los veinte vértices de un sólido regular dodecaedro son etiquetados como veinte famosas ciudades del mundo. El objetivo es ir alrededor del mundo pasando por cada ciudad una y solo una vez. La Figura 4.2 muestra una gráfica asociada con el problema. La solución es llamada circuito hamiltoniano. Sir Hamilton pasó por serias dificultades financieras para comprobar este juego.

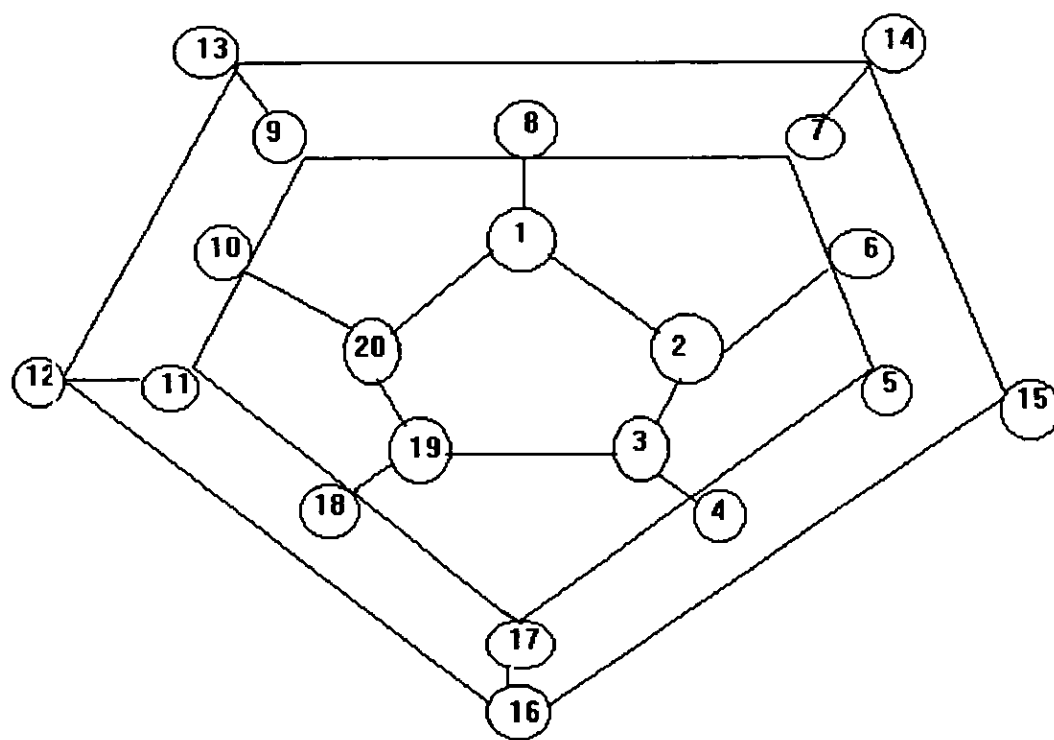


Figura 4.2 Problema de Hamilton.

Las siguientes contribuciones relevantes a la teoría de gráficas son probablemente aquellas de Kuratowski. El estableció el criterio que caracteriza la planariedad de gráficas.

En recientes años una enorme cantidad de investigaciones han sido conducidas en teoría de gráficas. Hay campos de interés humano que en alguna forma no puedan usarse los conceptos de teoría de gráficas, para resolver estos problemas, o al menos para proporcionar un mejor entendimiento de ellos. Aunque la teoría de gráficas es desarrollada rápidamente, pequeños intentos han sido hechos para estandarizar la nomenclatura usada para describir las gráficas.

GRÁFICAS.

DEFINICIÓN.

Una gráfica $G (V , E)$ es una estructura, la cual consiste en un conjunto de vértices $V = (v_1 , v_2 , \dots)$ y un conjunto de ramas $E = (e_1 , e_2 , \dots)$; cada rama incide en los elementos de una pareja de vértices (v_i , v_j) , no necesariamente ordenados o distintos.

Las gráficas son simples diagramas que consisten de puntos (vértices) y líneas (ramas), éstos diagramas o gráficas son usados para representar la forma de un sistema.

Las gráficas son simples abstracciones de la realidad, en este sentido las gráficas son modelos de sistemas, no cualquier sistema puede ser representado como una gráfica.

Como una regla general cualquier sistema que involucre relaciones binarias puede ser representado en forma de una gráfica.

El diagrama de la Figura 4.3 muestra una gráfica en un espacio de dos dimensiones. Una gráfica, sin embargo, no es restringida a un espacio de dos dimensiones.

El concepto de gráfica será introducido en este trabajo en un espacio Euclidiano.

Una gráfica está constituida por una colección de puntos y curvas simples con ciertas propiedades.

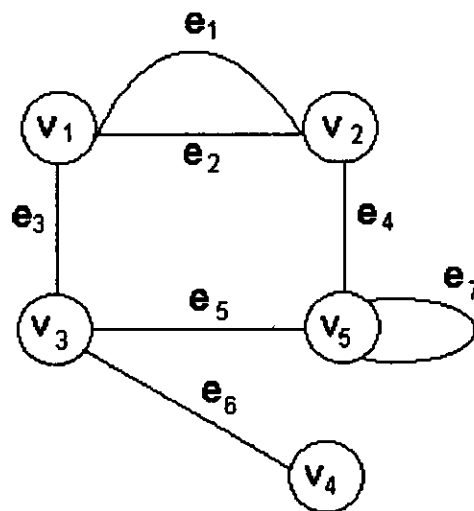
El conjunto de puntos asociados con una gráfica es llamado los vértices de una gráfica. Este conjunto será denotado por V con elementos $(v_1, v_2, \dots, v_N)$.

Una curva simple es una curva continua que no se intercepta ella misma.

Un conjunto de curvas simples es llamado ramas y es usado para indicar la relación entre vértices, y será denotado por E con elementos $(e_1, e_2, \dots, e_M)$.

Ejemplo:

En la Figura 4.3 tenemos representada la gráfica G .



$G(V, E)$

$V(v_1, v_2, v_3, v_4, v_5)$.

$E(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$.

Figura 4.3

Si el vértice v_1 y v_3 están relacionados la relación deberá ser indicada por una rama, la cual será designada por:

$$e_3 = (v_1 , v_3) \quad \text{ó} \quad e_3 = (v_3 , v_1)$$



Figura 4.4

En este sentido, la rama es una relación entre un par de vértices y la curva simple es únicamente una representación geométrica de la rama.

Por lo tanto, no tiene particular importancia, hacer énfasis en la longitud o forma de la curva.

Si los dos puntos extremos de la rama son diferentes, se dice que la rama es abierta como en la Figura 4.5 (a) de otra manera, es cerrada, Figura 4.5 (b).



(a)
Rama abierta



(b)
Rama cerrada

Figura 4.5

Podemos observar que en la gráfica $G (V , E)$ de la Figura 4.3 tenemos cinco vértices y siete ramas.

Como se definió, la gráficas $G (V , E)$ pueden ser no finitas, pero en este trabajo únicamente se tratará con gráficas finitas, es decir, V no vacío pero un número finito de vértices y posiblemente, cero número de ramas.

VÉRTICES.

■ **Vértices adyacentes.**

Se dice que dos vértices son **adyacentes** si están unidos por al menos una rama.

■ **Conjunto independiente de vértices.**

Un conjunto de vértices, tales que dos de ellos no son adyacentes, se dice que constituyen un **conjunto independiente de vértices**.

■ **Conjunto dominante de vértices.**

Un conjunto de vértices V_1 se dice que es un **conjunto dominante** si cualquier vértice de la gráfica no incluido en V_1 es adyacente al menos a un vértice incluido en V_1 .

■ **Grado de un vértice.**

El número de ramas que inciden en un vértice v (los rizados inciden dos veces) es llamado el **grado del vértice v** y se denota $\deg (v)$.

■ **Vértice aislado.**

Un **vértice aislado** es un vértice de grado, $d (v) = 0$.

■ **Vértice pendiente.**

Un **vértice pendiente** es un vértice con grado 1, $d (v) = 1$.

Ejemplo:

Podemos afirmar con seguridad que la Figura 4.6 representa un claro ejemplo de lo que es un vértice aislado, gráfica cuyo contenido se comprenderá mejor, apoyándonos en los conceptos y expresiones descritas anteriormente.

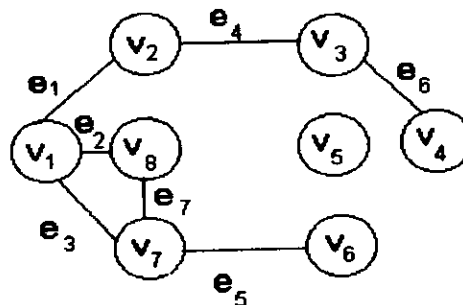


Figura 4.6, Gráfica con vértice aislado.

Los vértices v_1 y v_2 ; v_2 y v_3 ; v_3 y v_4 ; v_1 y v_8 , v_1 y v_7 ; v_7 y v_8 ; v_6 y v_7 son vértices adyacentes.

- El grado del vértice v_1 $d(v_1) = 3$.
- El grado del vértice v_2 $d(v_2) = 3$.
- El grado del vértice v_5 $d(v_5) = 0$.
(vértice aislado).
- El conjunto de vértices $V_1(v_1, v_3, v_5, v_6)$.
Forman un conjunto dominante.
- El conjunto de vértices $V_2(v_2, v_4, v_5, v_7)$.
Forman un conjunto dominante.
- El conjunto de vértices $V_3(v_5, v_3, v_8)$
Es un conjunto independiente.
- El grado de vértice v_6 ; $d(v_6) = 1$

■ **Vértice pendiente o aislado.**

Existe una relación interesante entre el grado y el número de vértices en una gráfica: el total de grados de los vértices de una gráfica es exactamente dos veces el número de ramas en una gráfica.

$$\sum_{v \in V} d(v) = 2 |E|$$

PROPIEDADES:

Una gráfica finita que tiene un número de vértices par tiene un grado impar.

■ **Gráfica k-regular.**

Asociado con cada gráfica $G (V, E)$:
Existe un vértice de grado mínimo:

$$\min \deg G$$

Y existe un vértice de grado máximo:

$$\max \deg G$$

Ejemplo:

La gráfica de la Figura 4.3
tiene:

$$\begin{aligned} \min \deg G &= 1 \\ &\text{y} \\ \max \deg G &= 4 \end{aligned}$$

Si en una gráfica:

$$\min \deg. G = \max \deg. G. = k$$

entonces a esta gráfica se le conoce cómo:

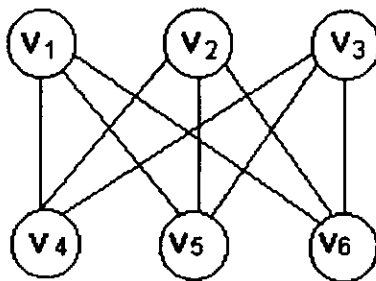
regular de grado K
o
 K -regular

Una gráfica regular de grado cero no tiene ramas

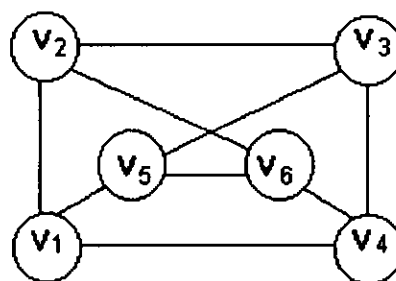
Si una gráfica es 2-regular entonces todos sus componentes son ciclos.

Ejemplo:

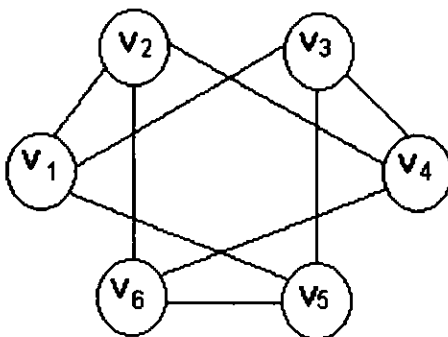
Las siguientes gráficas son 3-regular cada una con seis vértices y nueve ramas.



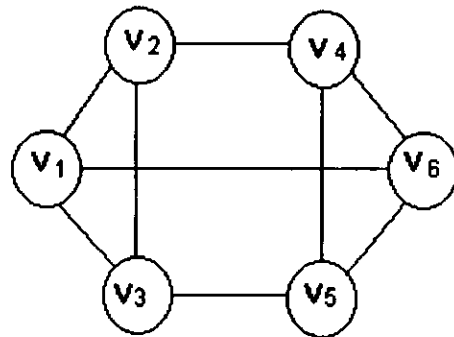
(a)



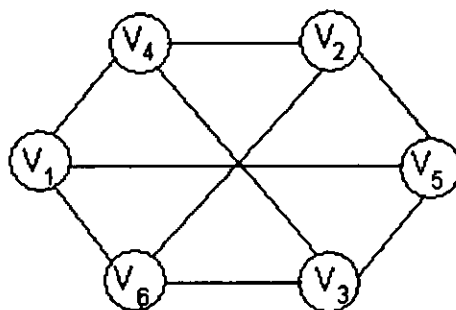
(b)



(c)



(d)



(e)

Figura 4.7

PROPIEDADES:

Para un grado impar K una gráfica regular tiene un número par de vértices.

RAMAS, SECUENCIA DE RAMAS Y CONECTIVIDAD.

■ Cadenas y circuitos.

En esta sección se estudian las características de las ramas. Las secuencias de las ramas definen rutas continuas dentro de una gráfica y son importantes en muchos problemas. Estas secuencias son llamadas **cadenas y circuitos**.

■ Gráficas degeneradas.

Una gráfica puede no tener ramas, es decir, consiste únicamente de un conjunto de vértices, tal gráfica se le conoce como **degenerada**.

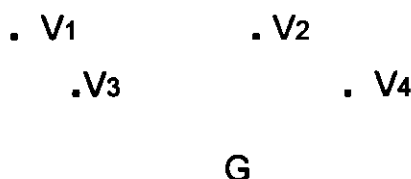


Figura 4.8, Gráfica $G (V, E)$; $E = 0$ degenerada.

■ Rizos (Loops).

Si una rama está asociada únicamente con un vértice, esto es, el inicio y el final de la rama son los mismos, entonces a esto se le conoce como un rizo (loop).

Ejemplo:



Figura 4.9, RIZO

■ Ramas paralelas.

Dos ramas que tienen los mismos puntos extremos son llamadas ramas paralelas.

Ejemplo:

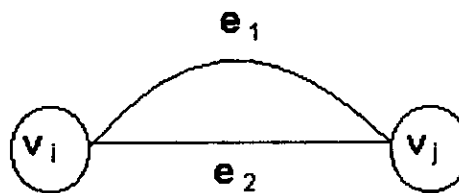


Figura 4.10 e_1 y e_2 son ramas paralelas.

■ Ramas adyacentes.

Las ramas que tienen al menos un vértice en común, se les conoce como ramas adyacentes.

Ejemplo:

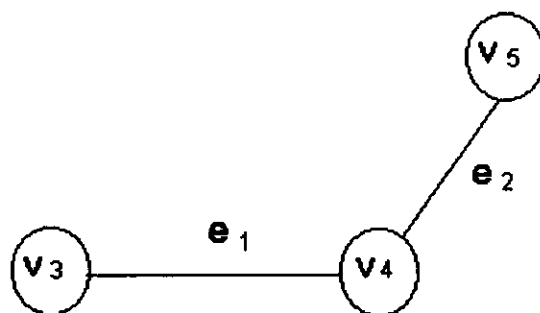


Figura 4.11 e_1 y e_2 son ramas adyacentes.

■ Secuencia de ramas y longitud.

En una gráfica $G (V , E)$, una secuencia de ramas entre v_0 y v_n es una **secuencia de ramas** adyacentes y no necesariamente distintas que van de v_0 a v_n . El número de tales ramas es llamado longitud de la secuencia de ramas.

Ejemplo:

En la Figura 4.6

Una secuencia de ramas es:

$\{ e_1, e_2, e_7, e_3, e_2 \}$ y su longitud es 5, nótese que las ramas en la secuencia no son únicas.

Otra secuencia sería:

$\{ e_1, e_4, e_6 \}$ con longitud 3, nótese que las ramas en la secuencia son únicas.

■ Secuencia cerrada y abierta.

Si $v_c = v_n$, la secuencia de ramas se dice que es **cerrada**. Si $v_0 \neq v_n$ se dice que la secuencia de ramas es **abierta**.

Ejemplo:

En la figura 4.7 (a);

Caso 1.

$\{ e_1, e_2, e_5, e_8, e_7, e_6, e_5, e_2 \}$

Es una secuencia cerrada de ramas en la gráfica G.

$\{ v_1, v_4, v_2, v_5, v_3, v_4, v_2, v_5, v_1 \}$

Es el conjunto de vértices asociados a la secuencia de ramas.

Observaciones:

a) Las ramas y los vértices pueden no ser únicos.

b) En el conjunto de vértices asociados.

$$v_p = v_q ; (v_1 = v_1)$$

Lo cual da la condición de secuencia cerrada.

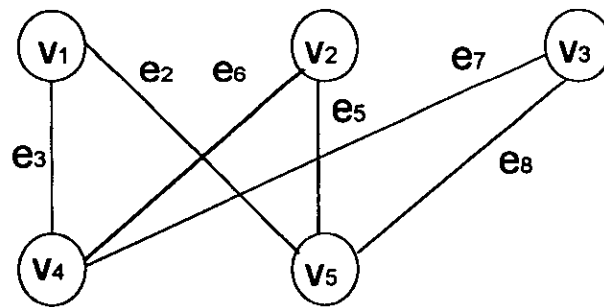


Figura 4.12

Caso 2.

$\{ e_3, e_6, e_5, e_8, e_7, e_6 \}$

Es una secuencia abierta de ramas en la gráfica G.

$\{ v_1, v_4, v_2, v_5, v_3, v_4, v_2 \}$

Es el conjunto de vértices asociados a la secuencia abierta de ramas en la gráfica G.

Observaciones:

a) Las ramas y los vértices de la secuencia pueden no ser únicos.

b) En el conjunto de vértices asociados:

$$V_p \neq V_q ; (V_1 \neq V_2)$$

Lo cual da la condición de secuencia abierta.

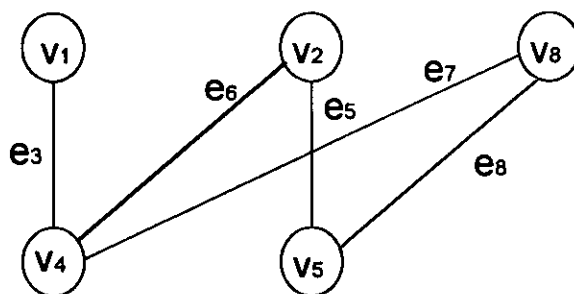


Figura 4.13 Secuencia abierta de ramas en la gráfica G.

■ CADENAS

Una secuencia de ramas abierta, en la cual las ramas son distintas se le conoce como **cadena**.

Los vértices de una cadena tienen las siguientes propiedades:

- a) Si v_o y v_n son los vértices inicial y final de una cadena, entonces éstos tienen grado uno.
- b) Cualquier otro vértice entre v_o y v_n tiene grado 2.

■ Cadena simple.

Una secuencia de ramas, en la cual los vértices (y por tanto las ramas) son distintos es llamada **cadena simple**.

Ejemplo:

En el ejemplo anterior para el caso 2. Si quitamos las ramas e_6 y e_7 , en la secuencia no existen ramas repetidas, es decir, es una cadena.

La secuencia (cadena) y el conjunto de vértices asociados queda como sigue.

$\{ e_3, e_4, e_5, e_8, e_7 \}$ y $\{ v_1, v_4, v_2, v_5, v_3 \}$

Observaciones:

a) En la cadena, las ramas son únicas, pero los vértices pueden ser repetidos.

Cuando los vértices son únicos, la cadena se denomina **cadena simple**.

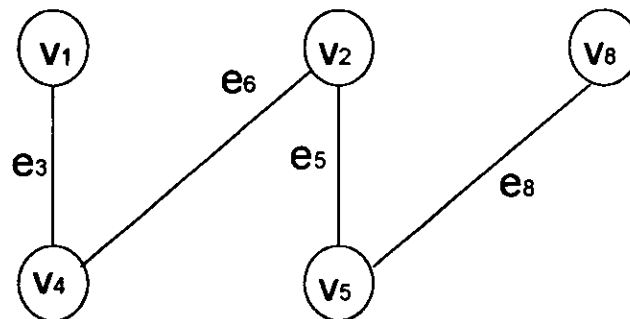


Figura 4.14 Cadena simple en la gráfica G.

b) $v_p = v_1$ tiene grado 1.

$v_q = v_8$ tiene grado 1.

Cualquier otro vértice tiene grado 2.

CIRCUITOS.

Una secuencia de ramas cerrada, con ramas distintas es llamada un **circuito**.

■ Circuitos simples.

Una secuencia cerrada de ramas, con $v_0 = v_n$ como los únicos vértices no distintos es llamado un **circuito simple**.

El circuito más corto posible en una gráfica simple y tiene una longitud de tres y se le conoce como triángulo.

Veamos la Figura 4.15 como un claro ejemplo:

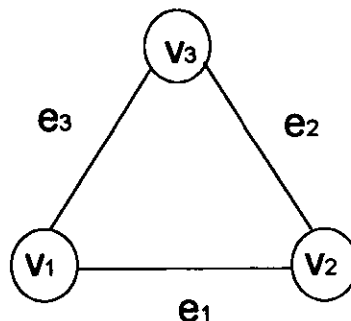


Figura 4.15 Circuito Triangular

En una gráfica general es posible tener un circuito de longitud uno o dos. El primero se llama **RIZO**, y el segundo **LUNA**.

El Rizo como ya lo mencionamos es una rama que inicia y termina en el mismo vértice. Las dos ramas que determinan una región se les denomina Lunas.

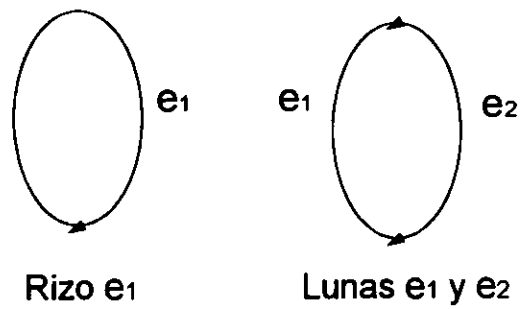


Figura 4.16

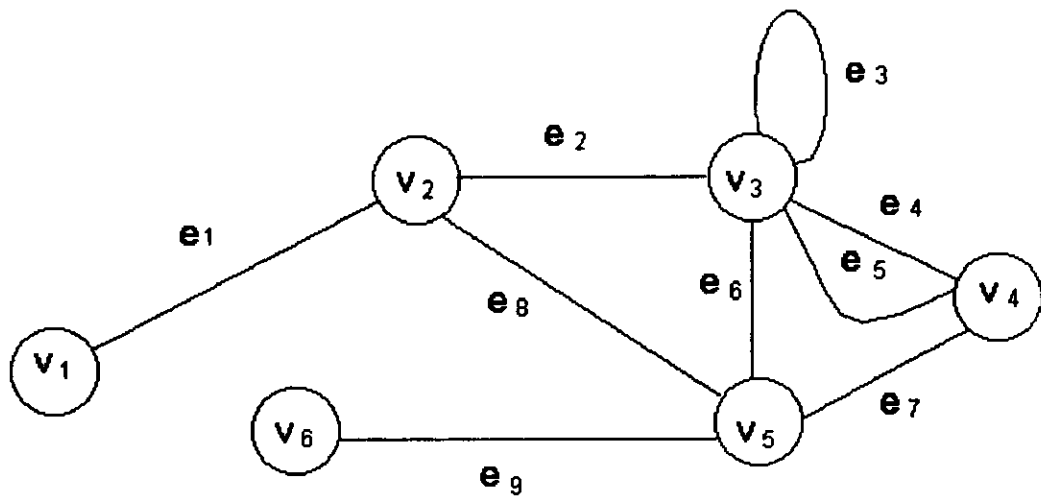


Figura 4.17

Ejemplo:

$$G = (E , V).$$

$$E = (e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9).$$

$$V = (v_1, v_2, v_3, v_4, v_5, v_6).$$

1) Cadena $(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ de v_1 a v_4 en G , con longitud 7 y vértices asociados $(v_1, v_2, v_3, v_3, v_4, v_3, v_5, v_4)$.

2) Cadena $(e_1, e_2, e_3, e_4, e_5, e_6)$ de v_1 , a v_3 en G , con longitud 6 y vértice asociados $(v_1, v_2, v_3, v_3, v_4, v_3)$.

3) Circuito simple (e_1, e_2, e_6) de v_1 a v_5 en G , con longitud 3 y vértices asociados (v_2, v_4, v_7, v_8) .

■ Gráfica conectada y no conectada.

Una gráfica G es **conectada** si cualquier par de sus vértices son unidos por al menos una cadena simple, de otra forma no es conectada.

Alternativamente, una gráfica $G (V , E)$ **no es conectada** si sus vértices en V pueden ser particionados en dos subconjuntos no vacíos V_1 y V_2 , tales que los vértices de V_1 , no son adyacentes a las vértices de V_2 .

■ Subgráficas.

Una subgráfica $G (V_1, E_1)$ de $G (V, E)$ es una gráfica con vértices en V_1 y ramas en E_1 , tales que $V_1 \subseteq V$ y $E_1 \subseteq E$.

■ Subgráfica propia.

Una subgráfica es **subgráfica propia** si G_1 y G no son idénticas.

Sea la gráfica de la Figura 4.18 que supone este tipo de subgráficas denotadas por las líneas punteadas y sus vértices:

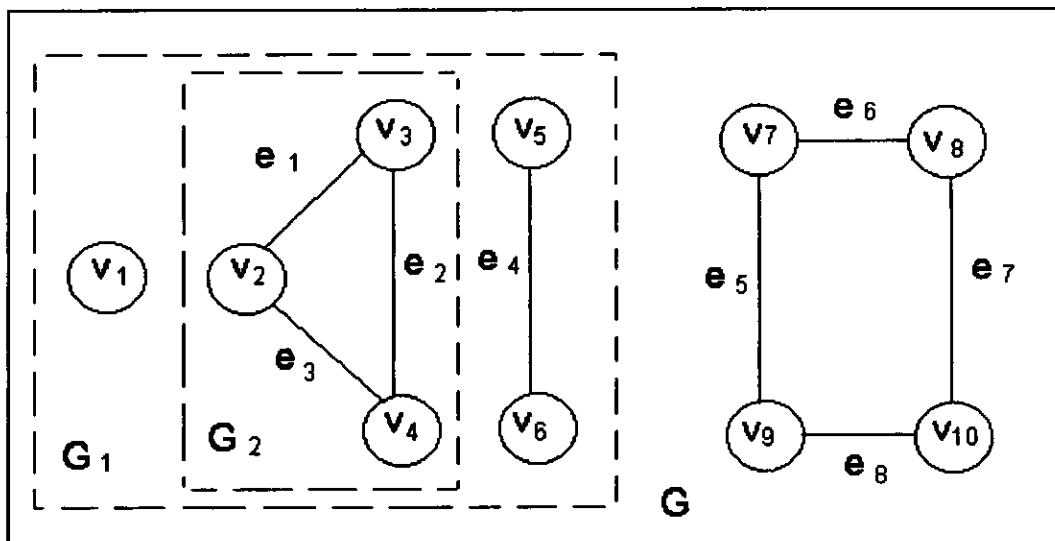


Figura 4.18

$G (V, E)$:

$V = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10})$.

$E = (e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8)$.

$G_1 (V_1, E_1)$:

$V_1 = (v_1, v_2, v_3, v_4)$.

$E_1 = (e_1, e_2, e_3, e_4)$.

$G_2 = (V_2, E_2)$.

$V_2 = (v_2, v_3, v_4)$.

$E_2 = (e_1, e_2, e_3)$.

Para $G_1 (V_1, E_1)$:

$$V_1 \subseteq V \text{ y } E_1 \subseteq E$$

entonces G_1 es una subgráfica de G .

Para $G_2 (V_2, E_2)$:

$$V_2 \subseteq V \text{ y } E_2 \subseteq E$$

entonces G_2 es una subgráfica de G .

Apoyados en los conceptos anteriores de **gráficas conectadas** y **subgráficas** podemos establecer una definición alterna de **circuito**.

■ **Circuito (definición alterna).**

Un **circuito** es una **subgráfica conectada** que cuando se extrae de la gráfica tiene vértices de grado 2.

■ **Supergráfica.**

Si G_1 es subgráfica de G , entonces se dice que G es una **supergráfica** de G_1 . En la Figura 4.19 se muestran una gráfica G y dos de sus subgráficas G_1 y G_2 .

Ejemplo:

G es supergráfica de G_1 y G_2 .

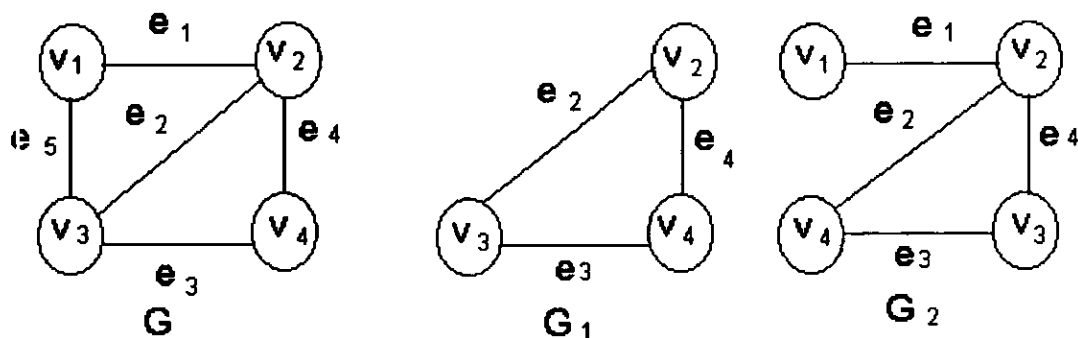


Figura 4.19 Gráfica G y dos Subgráficas G_1 y G_2 .

■ Gráfica conectada.

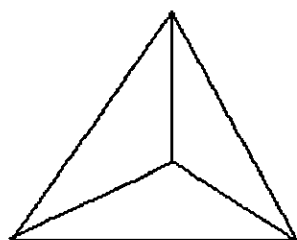
Una gráfica $G (V, E)$ se dice que es **conectada** si para dos vértices V_i, V_j existe una ruta de V_i a V_j .

PROPIEDADES :

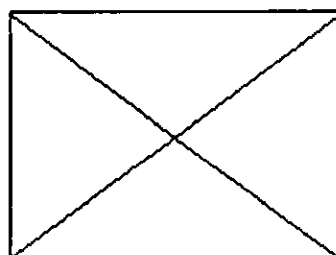
- a) Si una gráfica conectada contiene un ciclo, y si quitamos una arista, del ciclo, esta no desconecta a la gráfica.
- b) Si G es una gráfica conectada con p vértices y q aristas entonces:

$$p \leq q + 1$$

Ejemplos:



(a)



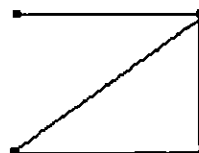
(b)



(c)



(d)



(e)

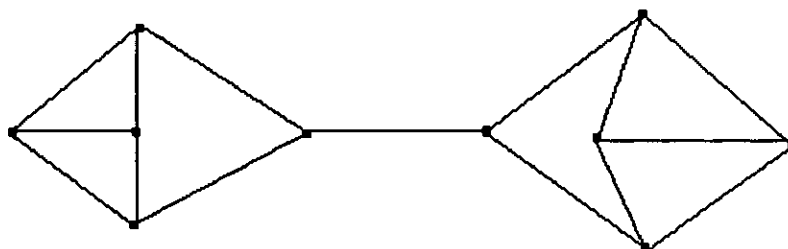
Figura 4.20

Las Figura 4.20 (a) y Figura 4.20 (b) pasan conectadas pero la Figura 4.20 (c) es no conectada; las Figuras 4.20 (d) y (e) no son conectadas.

■ **Gráfica mínimamente conexa (conectada).**

Una gráfica es mínimamente conexa si la eliminación de una cualquiera de sus ramas la hace desconexa.

Ejemplos:



a) La gráfica 3-regular de la Figura 4.21 es mínimamente conexa.

b) Las gráficas de la Figura 24 (a), Figura 24 (b) y Figura 24 (c) (árboles) son gráficas mínimamente conexas.

■ **Gráficas fuertemente conectadas.**

Si para cualquier par de vértices v_i y v_j hay una ruta de v_i a v_j , también hay una ruta de v_j a v_i .

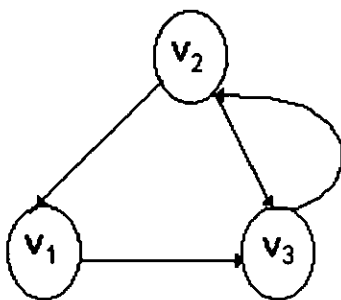


Figura 4.22

a) Si se tiene la ruta de v_i a v_2 (v_1, v_3, v_2).

También existe la ruta de v_2 a v_1 (v_2, v_1).

b) Para la ruta v_1 y v_3 .

(v_1, v_3)

También existe la ruta de v_3 a v_1 :

$$(v_3, v_2, v_1)$$

c) Para la ruta v_2, v_3

$$(v_2, v_3)$$

También existe la ruta de v_3 a v_2 :

$$(v_3, v_2)$$

■ **Componente.**

Componente de una gráfica es una subgráfica con conexión máxima.

Ejemplo:

En la gráfica G del ejemplo anterior tiene cuatro componentes.

$$G_1 (V_1, E_1):$$

$$V_1 = (v_1).$$

$$E_1 = \{\}$$

$$G_2 (V_2, E_2):$$

$$V_2 = (v_2, v_3, v_4).$$

$$E_2 = (e_1, e_2, e_3).$$

$$G_3 = (V_3, E_3).$$

$$V_3 = (v_5, v_6).$$

$$E_3 = (e_4).$$

$$G_4 = (V_4, E_4)$$

$$V_4 = (v_7, v_8, v_9, v_{10})$$

$E_4 = (e_5, e_6, e_7, e_8)$

■ **Gráfica completa.**

Una gráfica G en la cual cada par de vértices distintos están unidos por una arista, se llama gráfica completa.

Ejemplo:

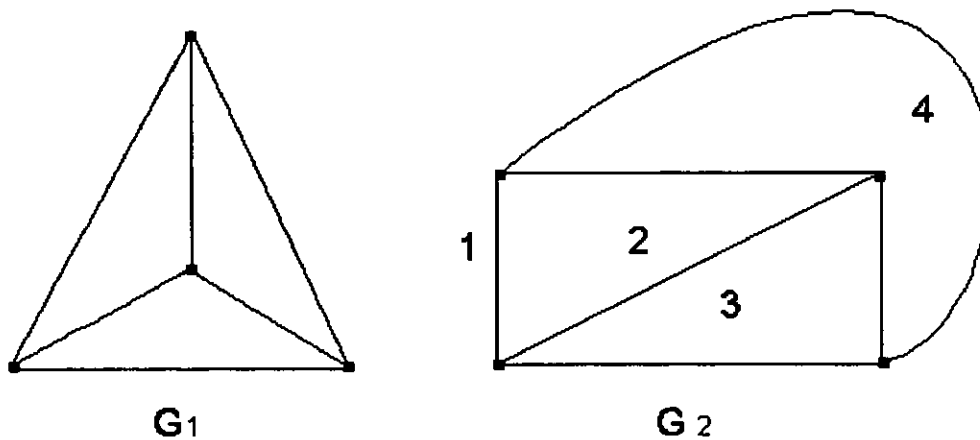


Figura 4.23 las gráficas G_1 y G_2 son gráficas completas de cuatro vértices.

Las gráficas de la Figura 4.23 son denotadas k_4 , en general, k_n es la gráfica con n vértices, y cualquier vértice es adyacente a cualquier otro vértice.

PROPIEDADES:

1) k_n tiene:

$$\binom{n}{2} = \frac{n(n-1)}{2}$$

Aristas

2) k_n tiene n subgráficas isomórficas, k_{n-1}

- **Gráfica complementaria.**

Una **gráfica complementaria** G_2 , de una gráfica simple G_1 es obtenida borrando las ramas de G_1 desde una gráfica completa que tiene los mismos vértices que G_1 . Así una gráfica complementaria de una gráfica completa es una gráfica sin las mismas ramas.

Ejemplo:

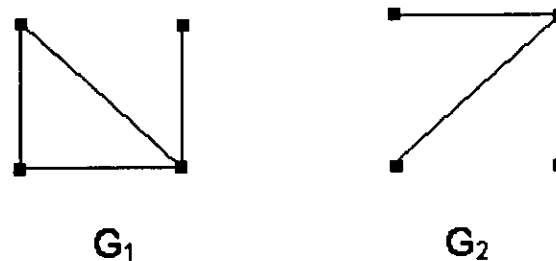


Figura 4.24 Gráficas complementarias G_1 y G_2 .

G_2 es una gráfica complementaria de G_1 .

- **Gráficas planas y no planas.**

Geométricamente podemos definir una gráfica plana como una gráfica que puede ser dibujada en una superficie tal que sus ramas se intercepten en sus vértices. De otra manera la gráfica no es plana.

Gráficas isomórficas.

Dos gráficas planas $G_1 (V_1, E_1)$ y $G_2 (V_2, E_2)$ se dice ser **isomórficas**, $G_1 = G_2$, si existe una correspondencia uno a uno entre sus vértices que preservan adyacencia.

En otras palabras las gráficas G_1 y G_2 con p vértices se dice que son **isomórficos** si los vértices de G_1 y G_2 ser etiquetados con los números de 1 a p siempre que el vértice v_G es adyacente al vértice V_i en G_1 , entonces el vértice V_i es adyacente al vértice v_j en G_2 e inversamente. Tal etiquetación es la misma como una correspondencia uno a uno entre $V (G_1)$ y $V (G_2)$ que preserva la adyacencia.

La Figura 4.23 muestra dos gráficas isomórficas lo mismo la gráfica de las Figuras 4.7 (a) a 4.7 (e).

REGIONES.

Cuando se dibuja una gráfica en un plano, ésta determina varias regiones. En la Figura 4.23 la gráfica G_2 determina cuatro regiones el número de regiones se denota por $|R|$.

■ Región exterior.

La región que no está limitada por las ramas de la gráfica se le conoce como **región exterior** o **región infinita** y está marcada con el número uno en la gráfica G_2 de la Figura 4.23.

Existe una importante relación entre el número de regiones $|R|$, el número de ramas $|E|$ y, el número de vértices $|V|$ y están dadas por las siguientes propiedades:

a) En una gráfica conectada sin circuitos se cumple la siguiente relación:

$$|V| = |E| + 1$$

b) Sea $G (V, E)$ una gráfica plana conectada con $|R|$ regiones, entonces:

$$|V| - |E| + |R| = 2$$

a esta relación se le denomina **Ecuación de Euler**.

■ Gráfica simple.

Una gráfica es **simple** si no tiene rizados ni ramas paralelas.

Ejemplo:

En la gráfica $G (E, V)$ de la Figura 4.17 si suprimimos las aristas e_3 (rizo) y e_5 (paralela a e_4), la gráfica G será una gráfica **simple**.

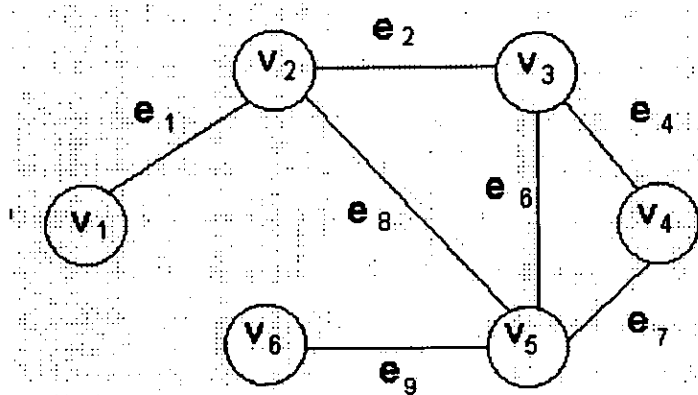


Figura 4.25 Gráfica simple.

PROPIEDADES:

Si una gráfica $G (V, E)$ es una gráfica simple plana conectada con $|V| \geq 3$, entonces:

$$3 |V| - 6 \geq |E|$$

Este resultado es importante porque proporciona un criterio para determinar si una gráfica es plana o no.

Si la desigualdad anterior no se cumple, entonces la gráfica considerada no es plana.

■ **Gráfica bipartita.**

Una gráfica $G (V, E)$ es **bipartita** si sus vértices pueden ser particionados en dos conjuntos V_1 y V_2 tales que cualquier rama en E tiene exactamente un extremo en V_1 y el otro extremo en V_2 .

■ **Gráfica bipartita completa.**

Una **gráfica bipartita completa**, es una gráfica completa $K_{m,n}$ y tiene $m+n$ vértices, las cuales las podemos separar en dos conjuntos, digamos, los vértices m son rojos, y los vértices n son azules. Cualquier vértice rojo es

adyacente a cualquier vértice azul y los vértices del mismo color no son adyacentes.

Ejemplo:

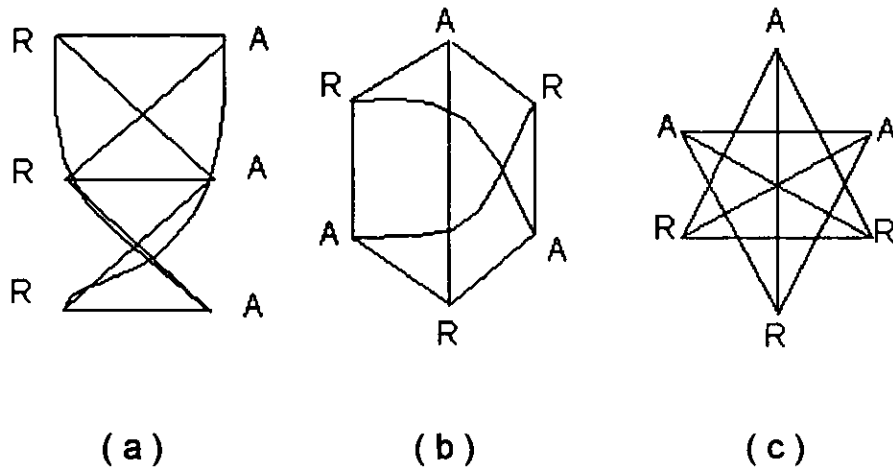


Figura 4.21 Gráficas completas bipartitas $k_{3,3}$

$$|V_1| = m \text{ y } |V_2| = n$$

GRÁFICAS DE KURATOWSKI.

Existen dos gráficas importantes que kuratowski las identificó como gráficas no planas. Estas gráficas se denominan como gráficas k_5 y $k_{3,3}$.

■ Gráfica k_5 de kuratowski.

Sea la gráfica k_5 una gráfica completa de cinco vértices, como se muestra en la Figura 4. 27, es una gráfica con un número pequeño de vértices que no es plana.

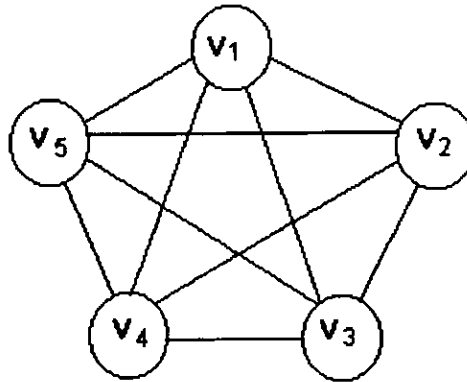
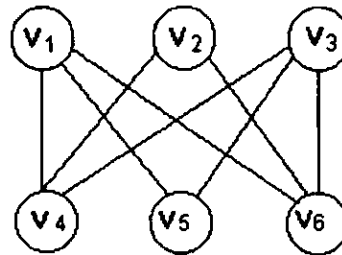


Figura 4. 27 Gráfica de kuratowski de cinco vértices (k_5)

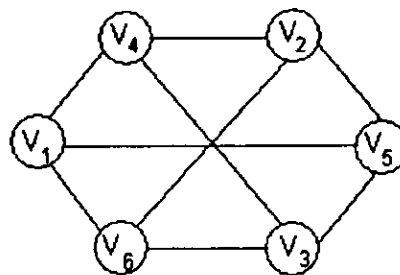
La segunda gráfica de kuratowski es:

Gráfica de kuratowski $k_{3,3}$.

Sea la gráfica $k_{3,3}$ completa bipartita de seis vértices como se muestra en la Figura 27 (a).



(a)



(b)

Figura 4.28 Gráfica de kuratowski $k_{3,3}$ con seis vértices y su gráfica equivalente isomórfica.

Ejemplo:

Resulta fácil establecer que K_5 y $K_{3,3}$ son no planas.

En K_5 , $|V| = 5$; $|E| = 10$

$$3|V| - 6 \geq |E|$$

$$3 \times 5 - 6 \leq |10|$$

$$9 \leq 10$$

No se cumple la relación, por lo tanto K_5 es no plana.

En $K_{3,3}$: es una gráfica bipartita, cualquier región está acotada por al menos cuatro ramas bipartitas. El máximo número de regiones es igual $2|E|/4$, cada rama puede ser frontera de dos regiones. De aquí:

$$R \leq 2|E|/4$$

o

$$|V| - |E| + R \leq |V| - |E| + 2|E|/4 = 6 - 9 + 4.5$$

o

$$|V| - |E| + |R| \leq 1.5$$

Esta ecuación no cumple con la relación $|V| - |E| + |R| = 1.5$ por lo tanto $K_{3,3}$ es no plana.

La importancia de las gráficas de Kuratowski radica en el hecho de que ayudan a caracterizar gráficas planas. Sin embargo, antes de presentar este resultado es necesario definir un nuevo término.

■ Gráficas Homeomórficas.

Dos gráficas son **homeomórficas** si alguna de ellas puede ser obtenida de la misma gráfica por una secuencia de operaciones que introducen secuencialmente un nuevo vértice para dividir una rama.

En la Figura 4.22 se muestran tres gráficas homeomórficas.

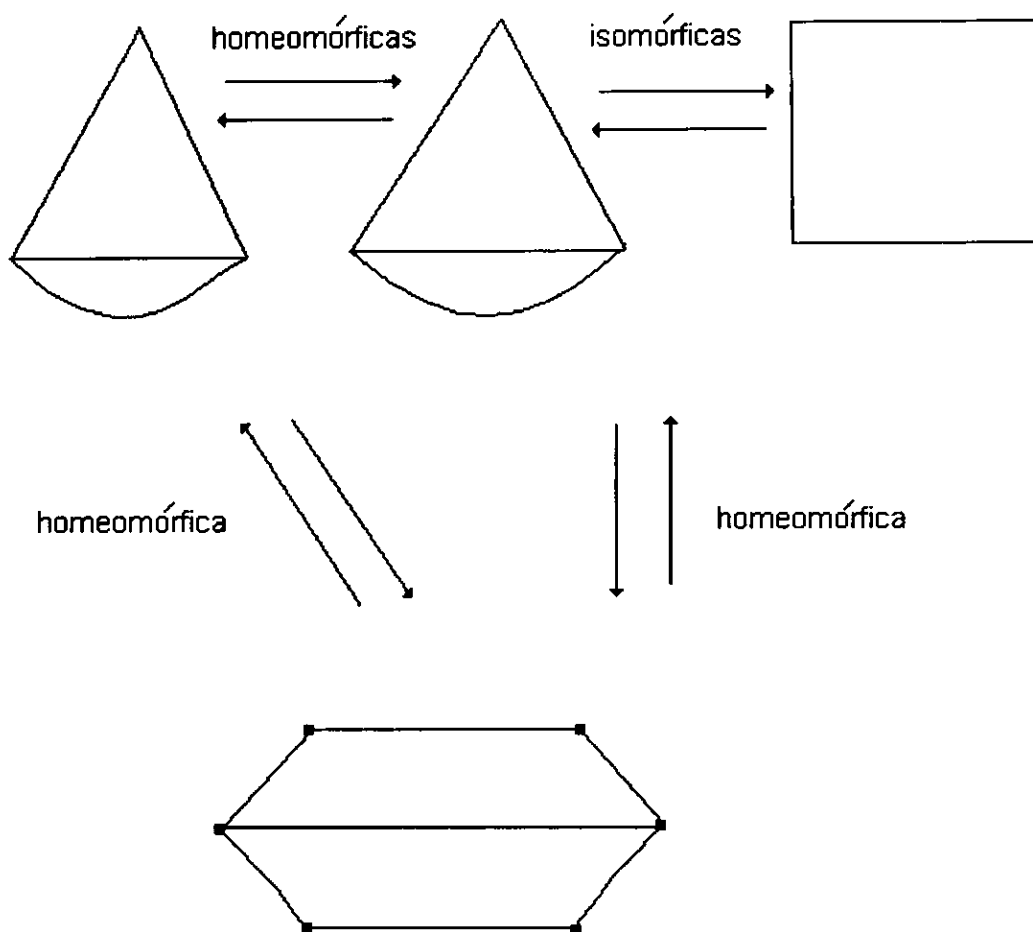


Figura 4.22 Ejemplos de gráficas homeomórficas

Teorema de Kuratowski.

Una gráfica es plana si y solamente si, no contiene una subgráfica homeomórfica K_5 o $K_{3,3}$.

GRÁFICAS SEPARABLES Y NO SEPARABLES.

Algunas veces es posible desconectar una gráfica conectada por substracción de un vértice.

■ Punto de articulación o punto de corte.

Un vértice v_i se dice ser el **punto de articulación** o **punto de corte** de una gráfica conectada G si la gráfica obtenida por substracción del vértice v y sus ramas asociadas son desconectadas.

En la Figura 4.30 los vértices v_4 , v_5 , v_6 y v_7 son todos puntos de articulación.

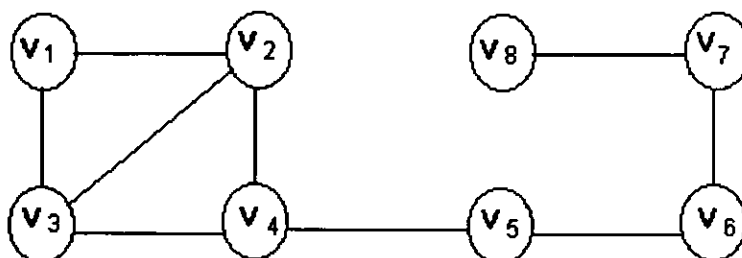


Figura 4.30 Gráfica que muestra puntos de articulación.

De lo anterior se desprende un resultado importante:

Un vértice v es un punto de articulación de una gráfica conectada si y solo así existen dos vértices, tales que el vértice v esta contenido en cualquier cadena y conecta esos dos vértices. Una gráfica que no tiene puntos de articulación es **no separable**, de otra forma es **separable**.

Ejemplo:

Las gráficas de la figura 4.28 no son separables.

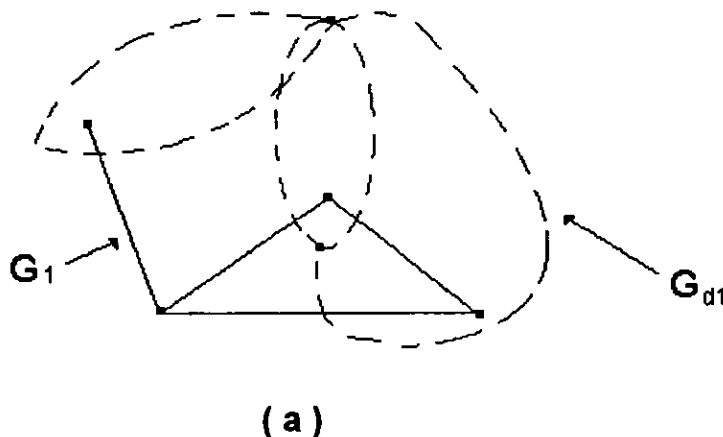
GRÁFICA DUAL.

Una gráfica dual $G_2 (V_2, E_2)$ de una gráfica $G_1 (V_1, E_1)$ es una gráfica tal en la cual el conjunto de vértices V_2 reemplaza las regiones o caras, delimitadas por los vértices del conjunto V_1 y sus aristas asociadas en G_1 y cumplen con lo siguiente.

- a) Los vértices del conjunto V_1 son conectados por líneas llamadas aristas.
- b) Las aristas de G_2 cortan a las de G_1 una sola vez.
- c) La gráfica G_1 es llamada primal.
- d) El dual de la gráfica dual será la gráfica primal.

En las Figuras 4.21 (a) y (b) tenemos dos gráficas G_1 y G_2 con sus respectivos gráficos duales G_{d1} y G_{d2} .

Ejemplos:



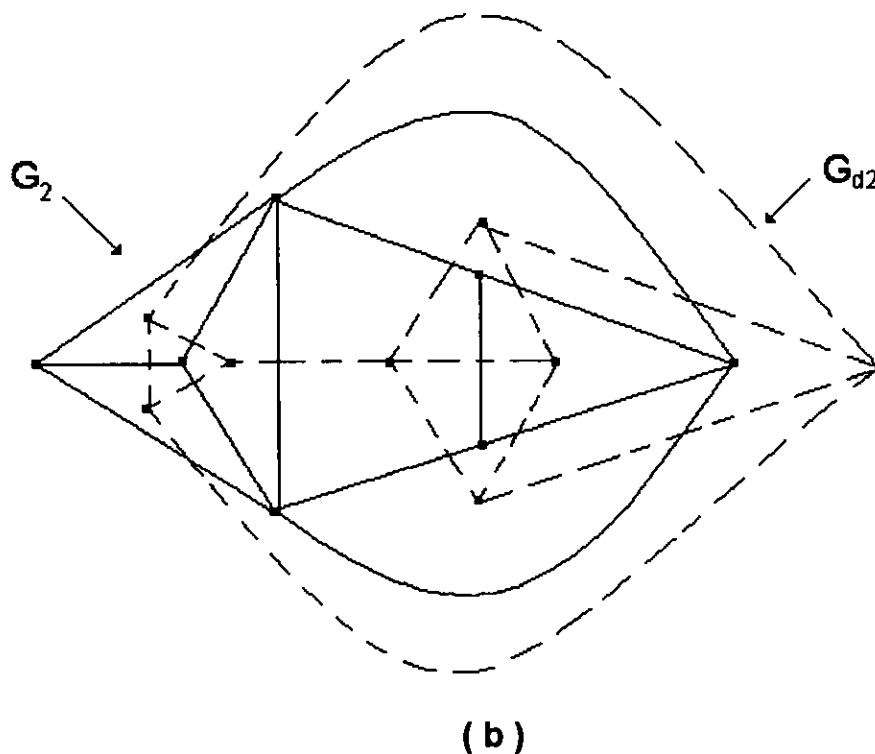


Figura 4.31 Gráfica G_2 y su gráfica dual G_{d2} .

GRÁFICAS DIRIGIDAS.

Una gráfica es **dirigida** si sus ramas tienen dirección.

■ **Arcos.**

A las ramas dirigidas se les denominan **arcos**.

Una gráfica dirigida se representa por $G (V, A)$ donde:

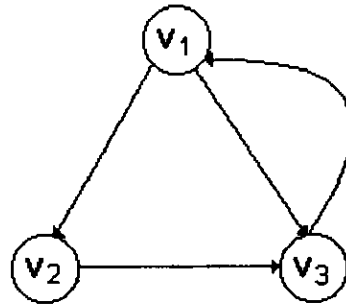
$V (v_1, v_2, \dots, v_m)$ conjunto de vértices de la gráfica G .

$A (a_1, a_2, \dots, a_n)$ conjunto de arcos de la gráfica G .

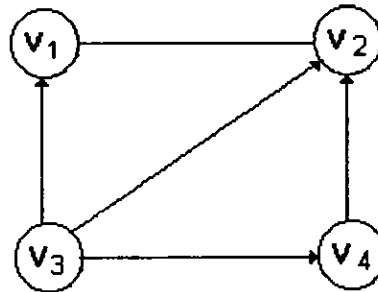
En el conjunto de arcos A , $a_1 = (v_1, v_2)$ será usado para representar un arco que se origina en el vértice v_1 y termina en v_2 . De una manera

análoga la expresión $a_2 = (v_2, v_1)$ ahora representan un arco que se origina en v_2 y termina en v_1 , por lo tanto, a_1 es diferente a a_2 , es decir (v_i, v_j) es una pareja ordenada.

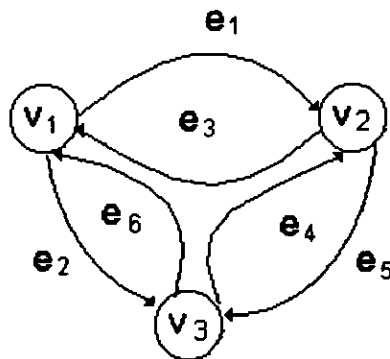
Ejemplos:



(a)



(b)



(c)

Figura 4.32 Ejemplos de gráficas dirigidas.

- **Arcos paralelos.**

Dos arcos que tienen los mismos puntos extremos son llamados **arcos paralelos**.

- **Arcos estrictamente paralelos.**

Dos arcos paralelos que tienen la misma orientación se dice que son **arcos estrictamente paralelos**.

- **Gráfica dirigida simple.**

Una gráfica dirigida es una **gráfica dirigida simple** si no tiene rizados ni arcos paralelos.

- **Arcos adyacentes.**

Dos arcos son **adyacentes** si tienen por lo menos un vértice en común.

- **Secuencia de arcos.**

Una **secuencia de arcos** entre v_p y v_q es un conjunto finito de arcos adyacentes, los cuales forman una secuencia desde v_p a v_q . Los arcos que representan esta secuencia no necesariamente son distintos.

- **Ruta.**

Una secuencia de arcos, con arcos distintos es llamada una **ruta**.

- **Ruta simple.**

Una ruta, en la cual todos los vértices son distintos es llamada una **ruta simple**.

- **Ciclo.**

Una ruta cerrada, esto es, una ruta que tiene $v_p = v_q$ es llamada un **ciclo**.

- **Ciclo simple.**

Si en un ciclo todos los vértices son distintos, excepto $v_p = v_q$, entonces es llamado un **ciclo simple**.

■ CIRCUITO DE EULER.

Un **circuito de Euler** en una gráfica G , es un circuito que contiene todas las ramas de G .

Ejemplo:

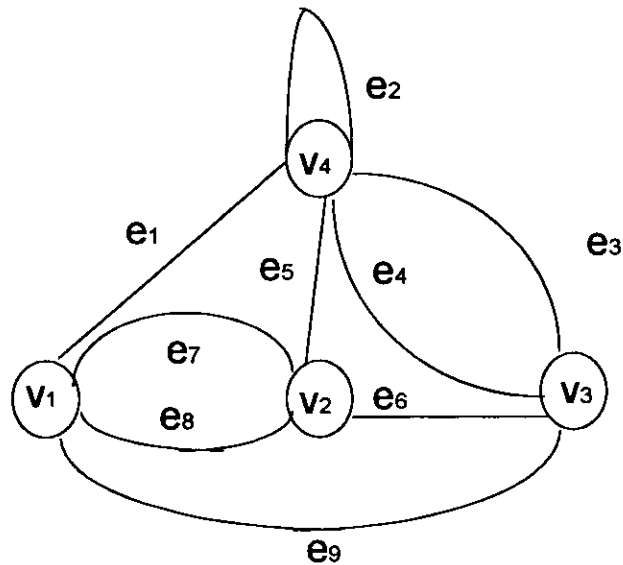


Figura 4.33

El circuito:

$$\{ e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9 \}$$

es un circuito de Euler.

Observaciones:

a) Este circuito no es único.

Ejemplo:

La gráfica siguiente denominada de Königsberg, no tiene circuito de Euler.

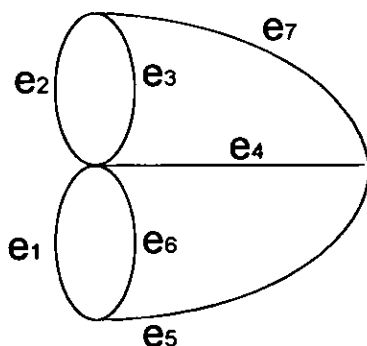


Figura 4.34 Gráfica de Königsberg.

PROPIEDADES DE LOS CIRCUITOS DE EULER:

- 1) Si una gráfica G tiene un circuito de Euler, entonces G es conectada y el grado de cualquier vértice es par.
- 2) Si una gráfica G es conectada y el grado de cualquier vértice de G es par, entonces G es un **circuito de Euler**.
- 3) Si todo vértice en una gráfica G tiene un grado positivo par, entonces cualquier vértice dado de G está en algún circuito de G .

■ Circuito de Hamilton.

Un **circuito de Hamilton** es un circuito simple en una gráfica G en el cual todos los vértices son visitados una y solo una vez.

■ Cadena de Hamilton.

Una cadena que incluye cualquier vértice de la gráfica exactamente una vez es llamada **Cadena de Hamilton**.

■ Ruta de Hamilton y Ciclo de Hamilton.

Ruta de Hamilton y Ciclo de Hamilton son términos que corresponden a gráficas dirigidas.

- **Gráfica cíclica.**

Una **gráfica cíclica** es una gráfica dirigida con al menos un ciclo.

- **Gráfica acíclica.**

Una gráfica que no tiene ciclos se le denomina **gráfica acíclica**.

- **Gráfica fuertemente conectada.**

Una gráfica se dice que es **fuertemente conectada** si para cualquier par de vértices v_i y v_j hay una ruta de v_i a v_j y una ruta de v_j a v_i .

- **Gráficas dirigidas isomórficas.**

Dos **gráficas dirigidas** son **isomórficas** si las gráficas correspondientes no dirigidas son isomórficas y los arcos correspondientes son orientados en la misma dirección.

Para caracterizar la incidencia de arcos es necesario definir dos términos:

- **Incidencia positiva e incidencia negativa.**

De la notación $a_1 = (v_1, v_2)$ donde significa que el arco a_1 incide de v_1 a v_2 . Se dice tener una **incidencia positiva** en v_1 y una **incidencia negativa** en v_2 .

- **Grado positivo y grado negativo de un vértice.**

El número de incidencias positivas y de incidencias negativas en un vértice v_i da el grado positivo y el grado negativo del vértice v_i . Y se denotan por:

$\text{pos } d(v_i) \text{ y } \text{neg } d(v_i) \text{ respectivamente}$

Así la relación entre las incidencias de una gráfica no dirigida y su correspondiente gráfica dirigida es la siguiente.

$$D(v) = \text{pos } d(v) + \text{neg } d(v) \text{ para todo: } v \in V$$

ÁRBOLES.

Un **árbol** es una gráfica $T (V_1 , E_1)$ no dirigida donde cualquier par de vértices de la gráfica es conectado exactamente por una cadena, esta afirmación implica dos puntos:

- a) Que la gráfica es conectada y que no tiene circuitos.
- b) La conectividad asegura que hay al menos una cadena entre dos vértices cualesquiera.

■ RAMAS.

Las aristas E_1 incluidas en T son llamadas **ramas** relativas a T .

■ CUERDAS.

Las aristas no incluidas en T son llamadas las **CUERDAS** relativas a T .

■ Árbol de expansión.

* Si el árbol T es tal que $V_1 = V$, esto es, todos los vértices de la gráfica también pertenecen al árbol, entonces se dice que el árbol es un **árbol de expansión** de la gráfica $G (V, E)$.

PROPIEDADES:

- a) No todas las gráficas tienen árbol de expansión. Un árbol es una gráfica mínimamente conectada y un árbol de expansión de una gráfica incluye todos los vértices de la gráfica.
- b) La existencia de un árbol de expansión implica que la gráfica es conectada. La inversa también es cierta.

Ejemplo:

En la Figura 4.35 muestra una gráfica con tres árboles de expansión diferentes.

Las cuerdas relativas a los árboles del ejemplo son mostradas con líneas punteadas.

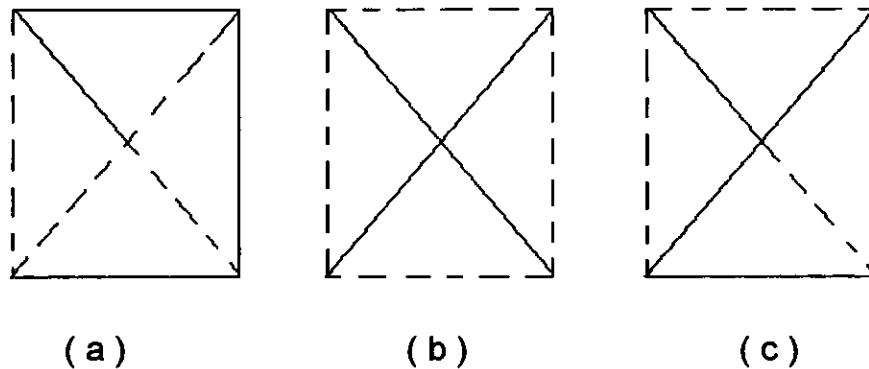


Figura 4.25 Árboles de expansión de una gráfica.

PROPIEDADES:

- a) Cualquier árbol contiene al menos dos vértices de grado uno.
- b) El número de ramas en el árbol es una menos que el número de vértices.

Esta relación siempre se cumple.

$$| V | = | E | + 1$$

- c) El concepto de árbol definido para gráficas puede ser extendido a gráficas dirigidas.
- d) Remover cualquier rama debe permitir dos vértices que no haya cadena entre ellos, esto es, hace una gráfica desconectada.
- e) Un árbol es una gráfica mínimamente conectada.
- f) Cualquier gráfica $G (V, E)$ contiene un árbol $T (V_1, E_1)$, tal que, T es subgráfica de G con:

$$v_1 \in V \text{ y } e_1 \in E$$

■ **Árbol dirigido, Raíz (v_0) .**

Un **árbol dirigido** $T (V, A)$ es una subgráfica dirigida si la correspondiente gráfica no dirigida es un árbol y si existe un vértice v_0 , tal que hay una recta de v_0 a todos los otros vértices incluidos en la subgráfica. Claramente este vértice es único.

-El vértice v_0 es llamado **raíz** del Árbol Dirigido.

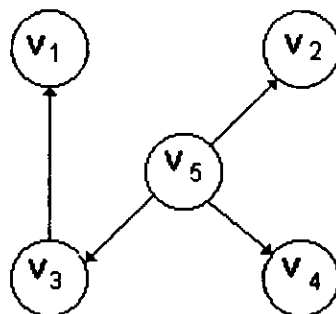
-Debido a que el árbol dirigido es una subgráfica mínimamente conectada la ruta de la raíz a cualquier otro vértice es necesariamente única.

Ejemplo:

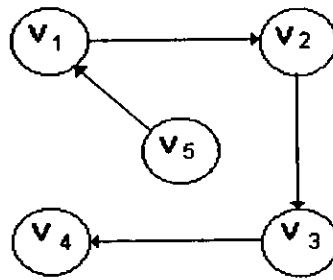
En la figura 4.36 se muestran dos Árboles Dirigidos Figuras (a) y (b) y una tercera gráfica Figura (c) que no es árbol dirigido.

Detalle:

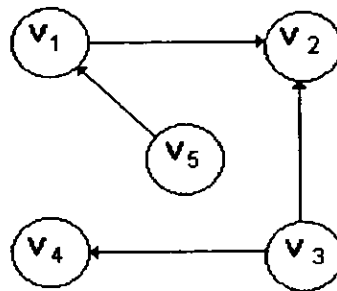
- En las Figuras 4.36 (a) y (b) se tienen dos árboles dirigidos cada uno tiene raíz en el vértice v_5 .
- En la Figura 4.36 (a), si la dirección del arco de v_5 a v_2 fuera cambiada de v_2 a v_5 entonces el arco tendría como raíz a v_2 .
- La Figura 4.36 (c) no representa un árbol dirigido porque no tiene raíz.



(a)



(b)



(c)

Figura 4.36 Ejemplos de árboles dirigidos (a y b, pero no c).

CONJUNTO DE CORTE.

Un conjunto de desconexión de una gráfica conectada $G (V, E)$ es un conjunto de ramas $E_1 \in E$ tal que después de sustraer de E_1 la gráfica residual $G_1 (V_1, E - E_1)$ ya no es conectada.

Un conjunto de desconexión divide necesariamente a la gráfica G en dos o más componentes.

El conjunto de vértices de estos componentes, V_1 y V_2 son tales que los vértices contenidos en V_1 ya no son adyacentes a aquellos contenidos en V_2

Si V_1 y V_2 son conjuntos no vacíos en G_1 entonces el conjunto de aristas E_2 que une a V_1 y V_2 forman un conjunto de desconexión. Tal conjunto es llamado **conjunto de corte**.

Ejemplo:

Sea la gráfica $G (V, E)$ como se presenta en la siguiente figura 4.37 (a):

Las líneas punteadas de la Figura 4.37 (b) muestran un **conjunto de desconexión** de la gráfica G .

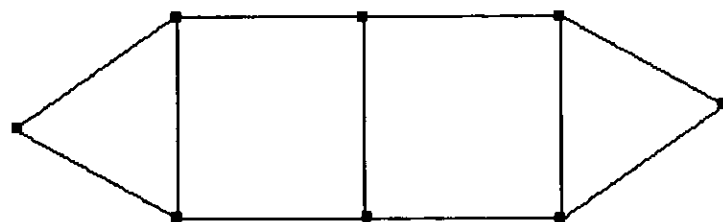
En la Figura 4.37, la gráfica muestra un conjunto de desconexión, un conjunto de corte y un conjunto de corte propio.

Nótese que la arista e_1 en la Figura 4.37 (b) deberá ser introducida en la gráfica sin alterar el número de componentes de la misma.

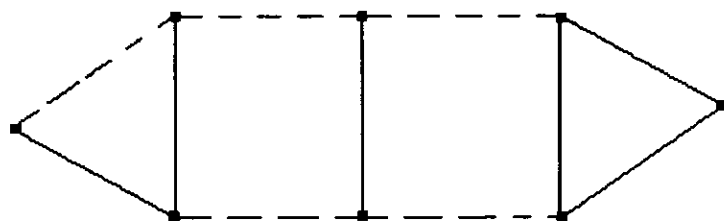
En la Figura 4.37 (c) se muestra el **conjunto de corte**.

En la Figura 4.37 (d) no hay aristas de la gráfica en la Figura 4.37 (a) que puedan ser introducidas sin reducir el número de componentes.

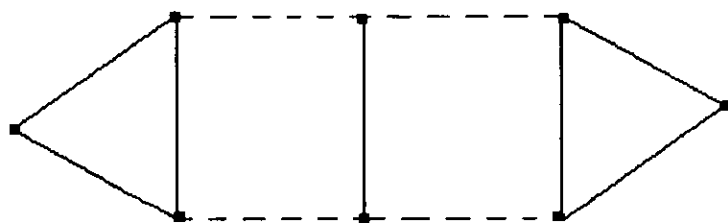
Las dos aristas representadas con líneas punteadas en la Figura 4.37 (d) también forman un conjunto de corte.



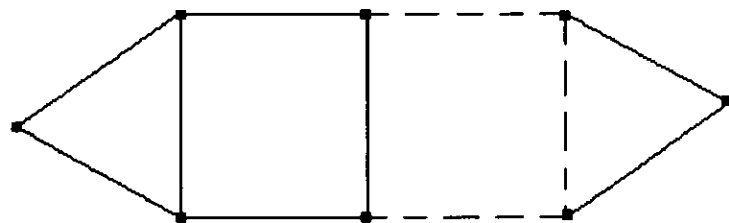
(a)



(b)



(c)



(d)

Figura 4.37

- **Conjunto de corte propio.**

Un conjunto de corte mínimo es llamado conjunto de **corte propio**.

PROPIEDADES:

- a) Si cualquier arista del conjunto de corte propio es reintroducida en la gráfica, la gráfica no deberá ser desconectada .
- b) Un conjunto de corte propio reduce a una gráfica con dos componentes exactamente.

La Figura 4.37 (d) muestra un conjunto de corte propio (hay muchos) en la gráfica en la Figura 4.37 (a).

MATRICES ASOCIADAS CON GRÁFICAS.

- **Matriz de Incidencias (B).**

Una matriz B es llamada **matriz de incidencia** de la gráfica $G = (V , E)$ si:

- 1) Cada renglón de B corresponde a un vértice de G.
- 2) Cada columna de B corresponde a una arista de G.
- 3) $b_{ij} = 1$ si la j-ésima arista incide en el i-ésimo vértice, de otra forma. $b_{ij} = 0$.
- 4) Las columnas correspondientes a los rizados son todas cero.

PROPIEDADES:

- 1.- La suma de todos los elementos de cada columna es igual a:

$$0 \bmod 2$$

Es decir la suma de todos los elementos de cualquier columna es igual a cero si representa un rizo, y es igual a dos si la columna representa una arista diferente a un rizo.

2.- La suma de los elementos de cualquier renglón de una matriz de incidencias es el grado del vértice correspondiente menos dos veces el numero de rizos que inciden en ese vértice.

3.- Si la gráfica G tiene n vértices y m aristas, entonces B es $n \times m$.

4.- Todas las aristas excepto los rizos inciden exactamente en dos vértices.

5.- Las columnas correspondientes de B tienen dos elementos con uno y $n-2$ elementos ceros.

6.- En el caso de gráficas dirigidas: para un arco a_k incidente de V_i a V_j b_{ik} toma el valor de $+1$ y b_{jk} el valor de -1 .

7.- La representación de una gráfica no es alterada si numeramos las columnas o renglones en diferente orden.

8.- Dos gráficas G_1 y G_2 son isomórficas si B_1 y B_2 son idénticas o pueden ser obtenidos por permutación de renglones o columnas.

9.- El rango de la matriz de incidencia B de una gráfica conectada G con un n vértices es $n-1$. Ejemplo: Sea la gráfica $G(V, E)$ como se muestra en la Figura 4.38.

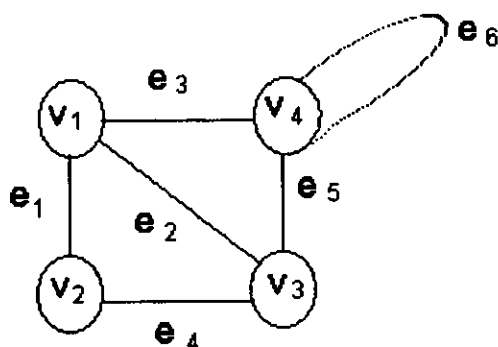


Figura 4.38

$$B = \begin{matrix} & & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & \text{grado} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} & \begin{matrix} 3 \\ 2 \\ 3 \\ 4-2=2 \end{matrix} \end{matrix}$$

■ Matriz de adyacencias (S).

Una representación matricial alternativa de una gráfica, describe una gráfica $G := (V, E)$ en términos de la adyacencia de todos los pares de vértices en G .

Una matriz S es llamada **matriz de adyacencias** de una gráfica $G := (V, E)$ o $G = (V, A)$ si:

1.- Los renglones y columnas de S corresponden a los vértices de G

y

2.- $S_{ij} = 1$ si la arista $(V_i, V_j) \in E$ [o arco $(V_i)(V_j) \in A$], $S_{ij} = 0$ de otra forma.

PROPIEDADES:

1) La matriz S es siempre en matriz cuadrada.

2) Los elementos en la diagonal de S representan los rizos.

3) El número de elementos diferentes de cero en S es menor o igual que el número total de arcos y rizos en una gráfica dirigida y el número de rizos más dos veces el número de aristas en una gráfica no dirigida.

4) Para una gráfica no dirigida la matriz de adyacencias es simétrica.

5) Para una gráfica dirigida la matriz, S la matriz de adyacencias no siempre es simétrica, en el caso de que fuese simétrica a la gráfica se le conoce como **gráfica dirigida simétrica**, en donde para

cualquier arco (v_i, v_j) hay un arco correspondiente (v_j, v_i) en dirección opuesta.

6) Siempre que haya aristas (o arcos) múltiples entre v_i, v_j de G , los elementos s_{ij} toman el valor del número de aristas diferentes en vez de uno.

■ Operaciones con gráficas.

Si consideramos dos gráficas $G = (V, E)$ y $G^* = (V, E^*)$ que tienen el mismo conjunto de vértices. V y matrices de adyacencia S y S^* asociados con ellos.

Unión (U) entre las gráficas G y G^* :

$$G \cup G^*$$

La matriz de adyacencias para la unión de G y G^* está dada por:

$$S + S^*$$

En el producto:

$$S \times S^*$$

los elementos de ésta matriz están dados por el número de cadenas distintas desde el vértice V_i al vértice V_j seleccionada de modo que de la primer arista $(V_i, V_k) \in E$, y la segunda arista $(V_k, V_j) \in E^*$.

$$L_{SS^*}$$

Los elementos de la matriz resultante están dados por:

$$(SS^*)_{ij} = \sum_k S_{ik} S^*_{kj}$$

En el caso particular donde $E = E^*$ y $G = G^*$ tendremos la multiplicación de S por sí misma y se denota:

$$S \times S = S^2$$

En general S^n a la n potencia indica la multiplicación de S por sí misma n veces.

PROPIEDADES:

Si S es la matriz de adyacencia de una gráfica G , cualquier elemento S_{ij} a la n potencia de la matriz de S a la n es igual al número de cadenas distintas de longitud n desde el vértice V_i al vértice V_j . Para una Gráfica Direccionada se obtienen rutas distintas.

■ Gráfica primitiva.

Se dice que una gráfica es **primitiva** si existe un entero m tal que hay una cadena de longitud m de cada vértice V_i a cualquier otro vértice V_j , esto es:

$$S_{ij}^n > 0 \text{ para todo } i \text{ y } j \text{ para alguna } n > 0.$$

ÍNDICE DE PRIMITIVIDAD.

Es el vértice m más pequeño que satisface la condición anterior. Como una regla general no es fácil encontrar el índice de primitividad.

A continuación se dan las condiciones para que una gráfica G sea primitiva.

Las condiciones necesarias y suficientes para que una gráfica dirigida (o no dirigida) sea primitiva son las siguientes.

- 1) La gráfica es fuertemente conectada y
- 2) El mayor común denominador de las longitudes de todos los ciclos (circuitos) simples es uno.

Ejemplo:

La matriz de adyacencias de la gráfica G es la figura como sigue:

$$S = \begin{matrix} & v_1 & v_2 & v_3 & v_4 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

■ Matriz C de circuitos.

Una matriz C es llamada **matriz de circuitos** de una gráfica. $G = (V, E)$ si:

- 1) Cada renglón en C corresponde a un circuito en G.
- 2) Cada columna de C corresponde a una arista en G .
- 3) $C_{ij} = 1$ si la arista j es incluida en el circuito i; de otra forma $C_{ij} = 0$.

PROPIEDADES:

- 1) Si la gráfica G tiene n vértices y m aristas. C tiene n vértices y m aristas. C tiene m columnas.
- 2) El número de renglones de C no es fácil de determinar porque una combinación lineal (unión mod. 2) de circuitos es también un circuito y tiene al menos un vértice en común.
- 3) Si una gráfica tiene más de un componente, entonces la matriz de circuitos correspondiente puede ser representable como una matriz de bloques diagonal.
- 4) Cualquier arista que incide en un vértice de grado uno (vértice colgante) no puede ser incluido en un circuito. Por lo tanto todos los elementos en la columna correspondiente a esta arista de C serán ceros.
- 5) La matriz de circuitos C de una gráfica G es ortogonal a la matriz B de incidencia de la misma gráfica, esto es:

$$BC = 0 \text{ mod } 2$$

Ejemplo:

Sea la matriz $G = (V, E)$

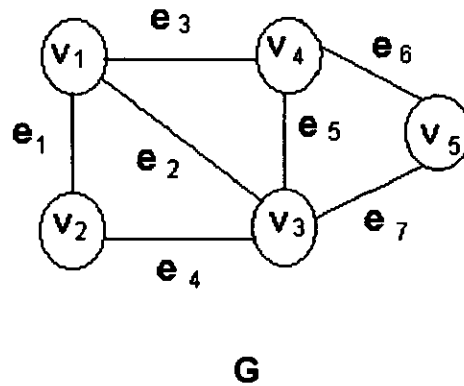
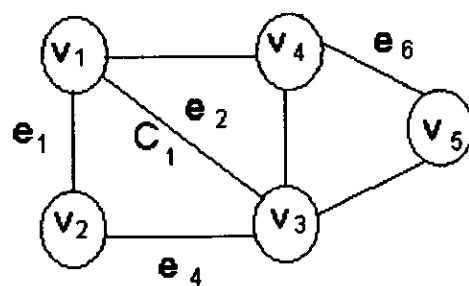


Figura 4.39 Gráfica G para ejemplo de matriz de circuitos.

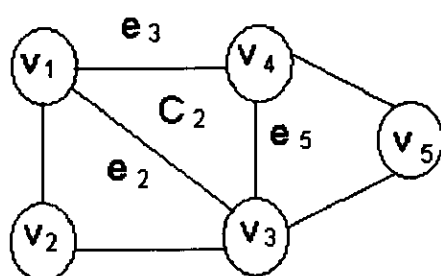
La matriz de circuitos es la siguiente:

$$C = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{matrix} \\ \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$



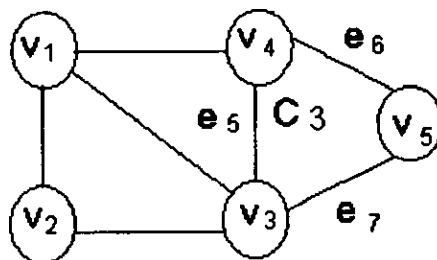
Circuito C_1

(a)



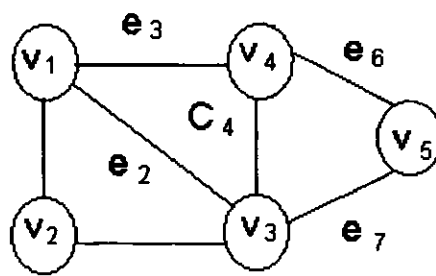
Circuito C_2

(b)



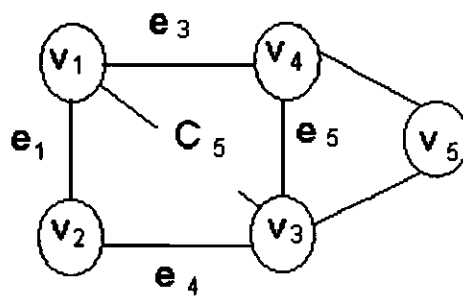
Circuito C_3

(c)



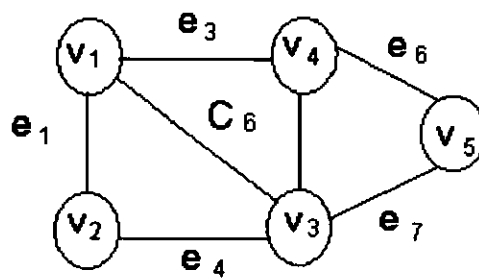
Circuito C_4

(d)



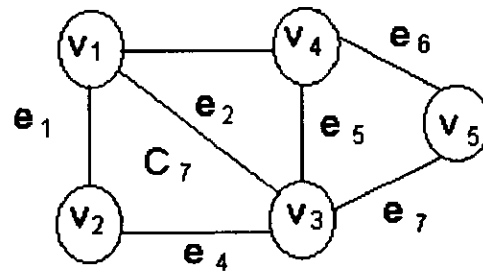
Circuito C_5

(e)



Circuito C_6

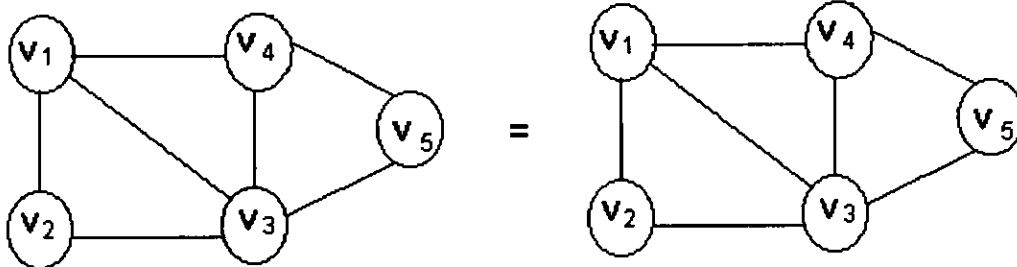
(f)



Circuito C_7
(g)

Es claro que:

$$C_4 = (C_2 + C_3) \bmod 2$$



C_4

$$= (C_2 + C_3) \bmod 2$$

(h)

Figura 4.40

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

Y la matriz de incidencias

$$B = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

$$BC' \bmod 2 = \begin{matrix} & B & \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} & \begin{matrix} C^+ \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix} \end{matrix}$$

Para el elemento $(BC)_{11}$:

$$(BC)_{11} = 1 \times 1 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 0 \times 0 + 0 \times 0 + 0 \times 0$$

$$(BC)_{11} = 1 + 1 + 0 + 0 + 0 + 0 + 0$$

$$(BC)_{11} = 0 + 0 + 0 + 0 + 0 + 0$$

$$(BC)_{11} = 0$$

Para el elemento $(BC)_{21}$

$$(BC)_{21} = 1 \times 1 + 0 \times 1 + 0 \times 0 + 1 \times 1 + 0 \times 0 + 0 \times 0 + 0 \times 0$$

$$(BC)_{21} = 1 + 0 + 0 + 1 + 0 + 0 + 0$$

$$= 1 + 1 + 0 + 0 + 0$$

$$(BC)_{21} = 0 + 0 + 0 + 0$$

$$(BC)_{21} = 0$$

.

.

.

Finalmente.

$$BC' = \begin{bmatrix} 0 \end{bmatrix}$$

- **Matriz C^+ de circuitos base.**

Como se vio anteriormente la unión de circuitos es también un circuito. Existe un conjunto mínimo de circuitos de una gráfica desde el cual, por medio de las uniones de dos o más circuitos podemos obtener todos los otros circuitos. A esta matriz se le conoce matriz de **Circuito Básico**.

- **Número ciclomático.**

El rango de la matriz C^+ de circuitos básicos se le conoce como **número ciclomático** de una gráfica.

PROPIEDADES:

- 1) Hay una correspondencia entre la matriz de circuitos básicos y las cuerdas de un árbol de la gráfica. Así para cualquier árbol de la gráfica, habrá una matriz de circuitos básicos correspondiente a las cuerdas del árbol.
- 2) Para escribir la matriz de circuitos básicos es conveniente colocar primero las aristas correspondientes a las cuerdas.
- 3) La matriz C^+ de circuitos básicos toma la forma :

$$C^+ = (I \ / \ C^-)$$

Esto es la primera parte de la matriz, es una matriz unitaria.

4) La matriz C^+ es importante para determinar la matriz K de conjunto de corte de la gráfica a partir de una matriz de circuitos.

Ejemplo:

Sean las gráficas $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ con árboles T_1 y T_2 como se muestra en las Figura 4.31 (a) y Figura 4.31 (b).

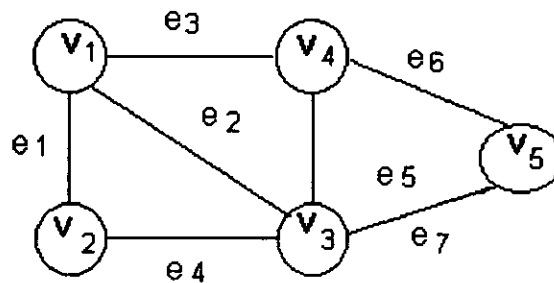


Figura 4.41 (a)

$$T_1 = \{ e_1, e_2, e_5, e_6 \}$$

$t_1 = \{ e_3, e_4, e_7 \}$ Conjunto de cuerdas.

La matriz C^+ de circuitos básicos queda para C_2 , como sigue:

$$C_1^+ = C_2 = C_3 = \begin{bmatrix} & e_3 & e_4 & e_7 & e_1 & e_2 & e_5 & e_6 \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{bmatrix}$$

$$C_1^+ = \left[\begin{array}{ccc|ccc} & & & & & & \\ & I & & & & & \\ & & & & C_i & & \end{array} \right]$$

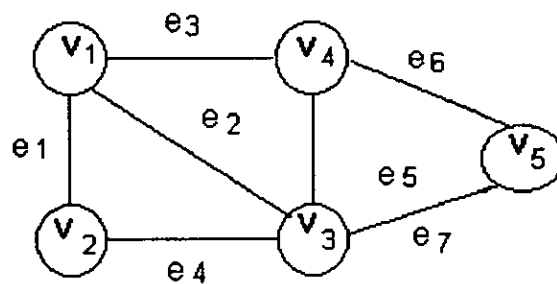


Figura 4.41 (b)

$$T_2 = \{ e_1, e_3, e_6, e_7 \}$$

$$t_2 = \{ e_2, e_4, e_5 \}$$

La matriz C_2 de circuitos básicos de G_2 queda como sigue:

$$C_2^+ = \begin{matrix} & e_2 & e_4 & e_5 & e_1 & e_3 & e_6 & e_7 \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

$$C_2^+ = \left[\begin{array}{ccc|ccc} & & & & & & \\ & I & & & C_2^- & & \end{array} \right]$$

■ Matriz de Ruta (P).

Una matriz P es llamada **Matriz de Ruta** de una gráfica $G (V , A)$ si:

- 1) Cada renglón de P corresponde a una ruta simple en la gráfica G .
- 2) Cada columna de P corresponde a un arco de G , y
- 3) $P_{ij} = 1$ si el arco j es incluido en la ruta y ; $P_{ij} = 0$, de otra forma.

PROPIEDADES:

- 1) Una matriz de ruta se define usualmente para gráficas dirigidas.
- 2) Una ruta (y Matriz de Ruta) puede ser definida para una gráfica no dirigida reemplazando cada arista por dos arcos en direcciones opuestas.

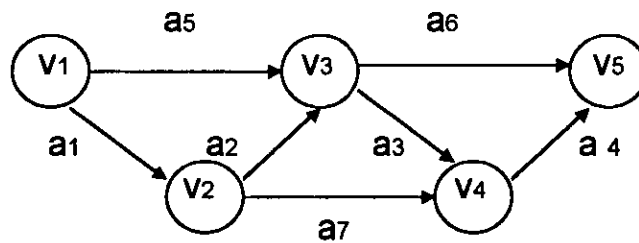


Figura 4.42 Gráfica para ejemplo de matriz de ruta.

$$P = \begin{matrix} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

■ Matriz conjunto de corte (K).

Una matriz K es llamada matriz de conjunto de corte de una gráfica $G = (V, E)$ si:

- 1) Cada renglón de K corresponde a un conjunto de corte G.
- 2) Cada columna de K corresponde a una arista de G, y
- 3) $K_{ij} = 1$ si la arista J es incluida en el conjunto de corte i. $K_{ij} = 0$, de otra forma.

PROPIEDADES:

- 1) Si la gráfica G tiene n vértices y m aristas K tiene m columnas.
- 2) El número de renglones de K no excede normalmente al número de vértices $|V|$, por lo tanto solo se consideran conjuntos de cortes básicos.
- 3) Si una gráfica G es conectada y no tiene puntos de corte, esto es, es z -regular, la matriz de corte K incluye la matriz de incidencias A .
- 4) La matriz de cortes K es ortogonal a la matriz de circuitos C esto es:

$$K C' = 0 \text{ mod } 2$$

Se asume que las aristas en las dos matrices son arregladas en el mismo orden.

- 5) El conjunto de corte divide a la gráfica en dos partes y se pueden tener dos casos:
 - A) No divide al circuito (El circuito y el conjunto de corte no tienen aristas en común).
 - B) Si divide al circuito (El conjunto de corte y el circuito tienen un número par de aristas para desconectar a la gráfica que esta en el circuito).

En ambos casos:

$$K C' = 0 \text{ mod } 2.$$

■ **Matriz del conjunto de corte básico K^+ .**

Es una submatriz de la matriz de conjunto de corte básico K , tal que los conjuntos de corte incluidos en la matriz K^+ son linealmente independientes.

PROPIEDADES:

- 1) El rango de la matriz de conjunto de corte básico K^+ es el mismo que el de K .
- 2) Hay una correspondencia entre el árbol de la gráfica y la matriz de conjunto de cortes. Un árbol es una gráfica mínimamente conectada; al remover una rama del árbol, desconecta el árbol en dos componentes. Una rama del árbol más cualquier cuerda adyacente a ambos componentes forma un conjunto de corte. El conjunto de corte, tiene una correspondencia en cada rama del árbol, y forman la matriz de conjunto de corte.
- 3) La matriz de corte básica tiene la forma

$$K^+ = (I / K^-)$$

- 4) K^+ es una matriz ortogonal con respecto a C .

$$K C' = 0 \text{ mod } 2$$

- 5) En base a la propiedad de ortogonalidad de K^+ y C^+ se establece un procedimiento para obtener la matriz de corte básico y viceversa.

$$K C' = 0 \text{ mod } 2$$

$$\begin{bmatrix} I & K^{-1} \end{bmatrix} \begin{bmatrix} C^{-1} \\ I \end{bmatrix} = 0 \pmod{2}$$

$$C^{-1} + K^{-1} = 0 \pmod{2}$$

$$C^{-1} = -K^{-1}$$

pero

$$-1 = 1 \pmod{2} \text{ y } -K^{-1} = K^{-1}$$

De donde

$$C^{-1} = K^{-1}$$

Conociendo C' obtenemos K' y viceversa.

■ Matriz de distancia L.

Aunque un arco en una gráfica es definido únicamente en términos de adyacencia y correspondencia entre pares de vértices, en algunos problemas llega a ser necesario definir cierta magnitud asociada con cada arco. Esta magnitud puede ser una distancia geográfica, una longitud geométrica ó un costo económico ó tiempo. Cuando ésta magnitud es definida por cada arco, es conveniente definir una **Matriz de Distancia**.

Una matriz L es llamada una **Matriz de Distancia** para una gráfica $G := (V, A)$ si:

- 1) Cada renglón de L corresponde a un vértice de la gráfica G , y
- 2) Cada columna de L corresponde a un vértice de la gráfica G , y
- 3) l_{ij} es la magnitud definida para el arco (v_i, v_j) .

PROPIEDADES:

- 1) Se considera que l_{ij} es no negativa.
- 2) La distancia entre vértices no adyacentes considera infinita.
- 3) La distancia de una ruta es tomada como la suma de la distancia de todos los arcos incluidos en la ruta.
- 4) Si hay arcos paralelos se considera únicamente la distancia mas larga o la mas corta, dependiendo del problema a resolver.

Ejemplo:

Sea la gráfica $G (V, A)$ mostrada en la siguiente figura:

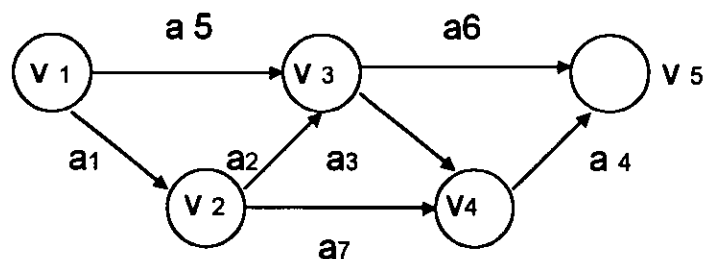


Figura 4.43

Recordemos que el termino longitud se usa para designar el número de arcos en una ruta.

Pero en este caso consideramos distancia como la suma de las magnitudes asociadas con cada arco en la ruta.

La matriz de distancias para esta gráfica es:

$$L = \begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{bmatrix} 0 & 12 & 6 & \infty & \infty \\ \infty & 0 & 8 & 10 & \infty \\ \infty & \infty & 0 & 6 & 8 \\ \infty & \infty & \infty & 0 & 7 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix} \end{matrix}$$

Considerando las rutas P. y Pu de la matriz de rutas sus distancias serán:

$$P_1 = 6 + 8 = 14$$

$$P_4 = 12 + 8 + 6 + 7 = 33$$

Ejemplos:

Como se estableció anteriormente las matrices asociadas con gráficas son generalmente matrices binarias (ceros, unos).

Se emplean dos tipos de aritmética para la manipulación de matrices binarias. Estas son módulo 2 (o mod 2) y la aritmética Boollane.

El resultado para cualquier operación en mod 2 es obtenido dividiendo el resultado de la aritmética normal por 2 y tomando únicamente la porción remanente de la división ignorando el cociente. En base a ésta definición las reglas para mod 2 son:

$$1 \times 1 = 1 \quad 1 + 1 = 0$$

$$1 \times 0 = 0 \quad 1 + 1 = 1$$

$$0 \times 0 = 0 \quad 0 + 0 = 0$$

Reglas para mod 2.

En la aritmética Booleana el operador $+$ puede ser considerado como la representación de la unión en el álgebra de conjuntos y el operador $\times$ como la intersección de 2 conjuntos las reglas para la aritmética Booleana están dadas por:

$$1 \times 1 = 1 \quad 1 + 1 = 1$$

$$1 \times 0 = 0 \quad 1 + 0 = 1$$

$$0 \times 0 = 0 \quad 0 + 0 = 0$$

Reglas para la aritmética Booleana.

Nótase que irremediabilmente hay una diferencia entre los resultados de la aritmética mod 2. y la aritmética Booleana y es en la operación $1+1$.

CAPÍTULO 5

ANÁLISIS DEL MEBC.

En este capítulo se hace un estudio , apoyado en la Teoría de Gráficas de las relaciones relevantes del **MEBC** en las características de **EJES ESTRUCTURALES** y **ALCANCES DEL MODELO** considerando en este caso únicamente los puntos de Salida Terminal y de **Licenciatura**.

Empecemos con el estudio de los **Ejes estructurales**. Cuyos elementos son:

- 1) v_1 : Trinomio Educación-Sector Productivo-Sociedad..
- 2) v_2 : Currículum.
- 3) v_3 : Docencia.
- 4) v_4 : Cultura Organizacional Académica.

Para los **Alcances del modelo** consideramos:

- 5) v_5 : Salidas laterales.
- 6) v_6 : Licenciatura.

Como premisa para analizar las relaciones de los puntos anteriores, es necesario establecer los siguientes conceptos.

1) Contexto.

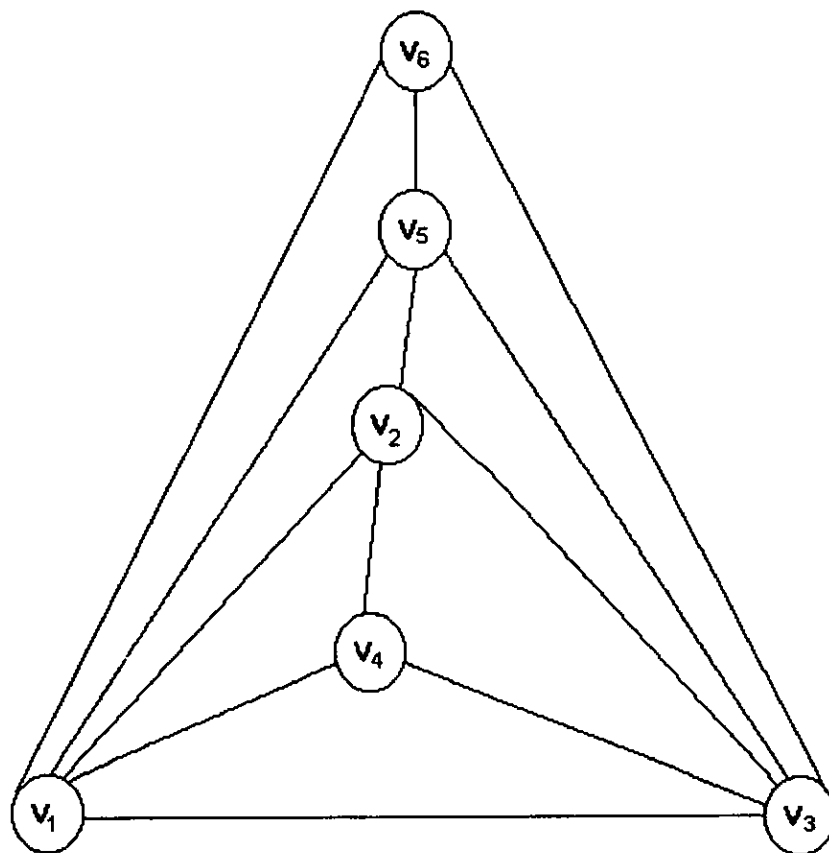
México como un país en crisis general, la cual se agudiza por la apertura comercial con países más ricos y desarrollados debe enfrentar el cambio reformando sus estructuras políticas económicas y sociales. La reforma educativa es uno de los pilares para lograr el desarrollo integral del país. Los modelos educativos que se han aplicado en México, no han sido del todo exitosos. Actualmente se pretende establecer en la educación técnica media superior y superior, el modelo relativamente nuevo (en México) , **MEBC**. Modelo probado con éxito en otros países (Canadá, Estados Unidos, Inglaterra, etc.) países ricos con economías, que rebasan a la mexicana.

Para implantar con éxito en México el MEBC, en un contexto como se describe al inicio muy diferente al de los países del primer mundo para los cuales ha funcionado satisfactoriamente es necesario hacerlo de una manera racional aplicando las Técnicas de la Investigación de Operaciones en particular la Teoría de Gráficas.

2) Objetivos:

- a) El estudiante termina una licenciatura.
- b) Carrera de técnico medio (salida lateral).
- c) El estudiante al terminar deba estar capacitado en competencias requeridas en la industria.

Los elementos considerados los podemos relacionar de la manera siguiente:



.Figura 5.1 Gráfica G de las relaciones de los ejes estructurales y alcances del modelo.

Las áreas de síntesis serán como se muestra en la Figura 5.2 marcadas con d_1 a d_8 :

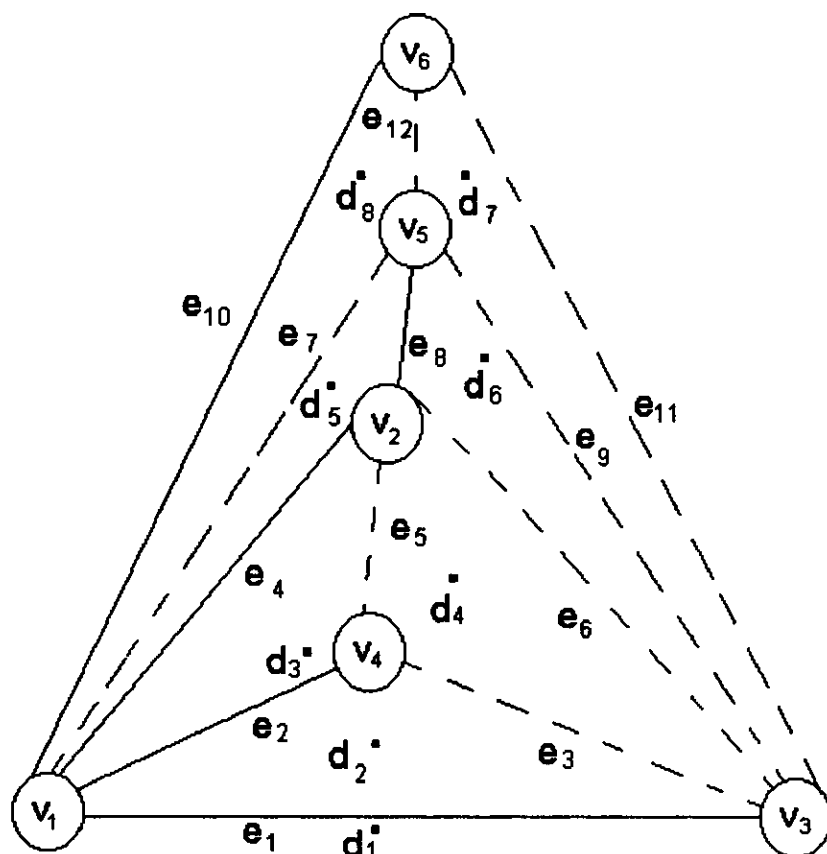


Figura 5.2 Gráfica G con las Áreas de Síntesis d.

Las relaciones anteriores las podemos representar matricialmente con la Matriz de Adyacencias como sigue:

	Trinomio E-SP-E	Curriculum	Docencia	Cultura Org. Ac.	Salidas Laterales	Licenciatura	Grados
	V1	V2	V3	V4	V5	V6	
V1	0	1	1	1	1	1	5
V2	1	0	1	1	1	0	4
V3	1	1	0	1	1	1	5
V4	1	1	1	0	0	0	3
V5	1	1	1	0	0	1	4
V6	1	0	1	0	1	0	3

$$\begin{aligned}
 \Sigma g &= 24 \\
 \text{N}^{\circ} \text{ de ramas} &= 12 \\
 \text{rango } r = n - 1 = 6 - 1 &= 5 \\
 \text{nulidad } M = e - (n - 1) &= 1
 \end{aligned}$$

Matriz de Adyacencias de la gráfica G.

En esta matriz como se estableció en el capítulo 4 :

- 1 Indica cierta relación entre dos vértices.
- 0 Indica relación nula entre dos vértices.

Se podría establecer la intensidad de la relación entre los vértices, débil, fuerte, indispensable, pero esto se considera como el proceso de síntesis. A continuación se representa esquemáticamente la gráfica G con su dual G_d .

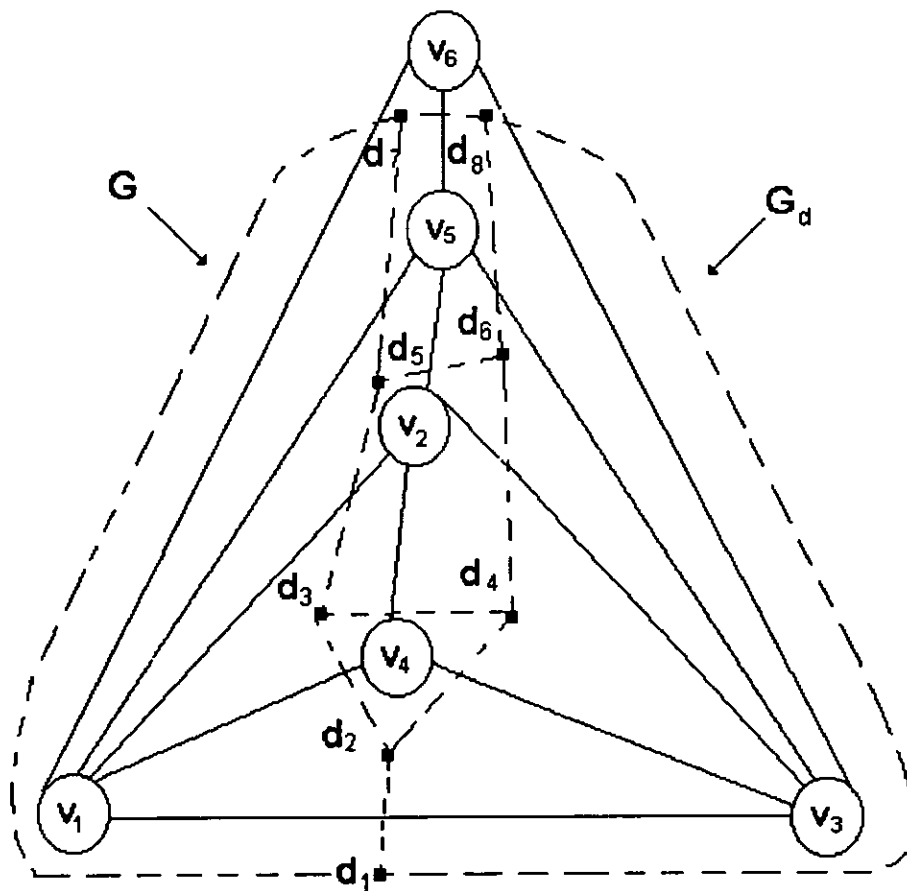


Figura 5.3 Gráfica Dual G_d de la Gráfica Primal G .

Los vértices d_1 a d_8 representan las regiones determinadas por G y relacionadas nos conforman la gráfica dual G_d estas regiones representan las Áreas de Síntesis.

La representación matricial de la gráfica dual G_d esta dada en la siguiente matriz.

	d ₁	d ₂	d ₃	d ₄	d ₅	d ₆	d ₇	d ₈	Grado
d ₁	0	1	0	0	0	0	1	1	3
d ₂	1	0	1	1	0	0	0	0	3
d ₃	0	1	0	1	1	0	0	0	3
d ₄	0	1	1	0	0	1	0	0	3
d ₅	0	0	1	0	0	1	0	1	3
d ₆	0	0	0	1	1	0	1	0	3
d ₇	1	0	0	0	0	1	0	1	3
d ₈	1	0	0	0	1	0	1	0	3

$$\Sigma g_d = 24$$

$$\text{ramas} = 12$$

$$\text{rango} = n - 1 = 7$$

$$\text{nulidad} = e_d - r_d = 5$$

Matriz de Adyacencias de la gráfica dual G_d .

Nótese que:

$$e = e_d = 12.$$

$$r = r_d (7 = 7).$$

$$\mu_d = \mu (5 = 5)$$

Para analizar las Áreas de Síntesis fundamentales, determinamos los Árboles de Expansión T y T_{d1} de las gráficas G primal figura 5.4 y G_d Dual figura G_d , 5.5 . La figura 5.6 representa la subgráfica M formada por cuerdas de T .

El Arbol de Expansi3n T de la gr3fica G quedar3 como en la Figura 5.3:

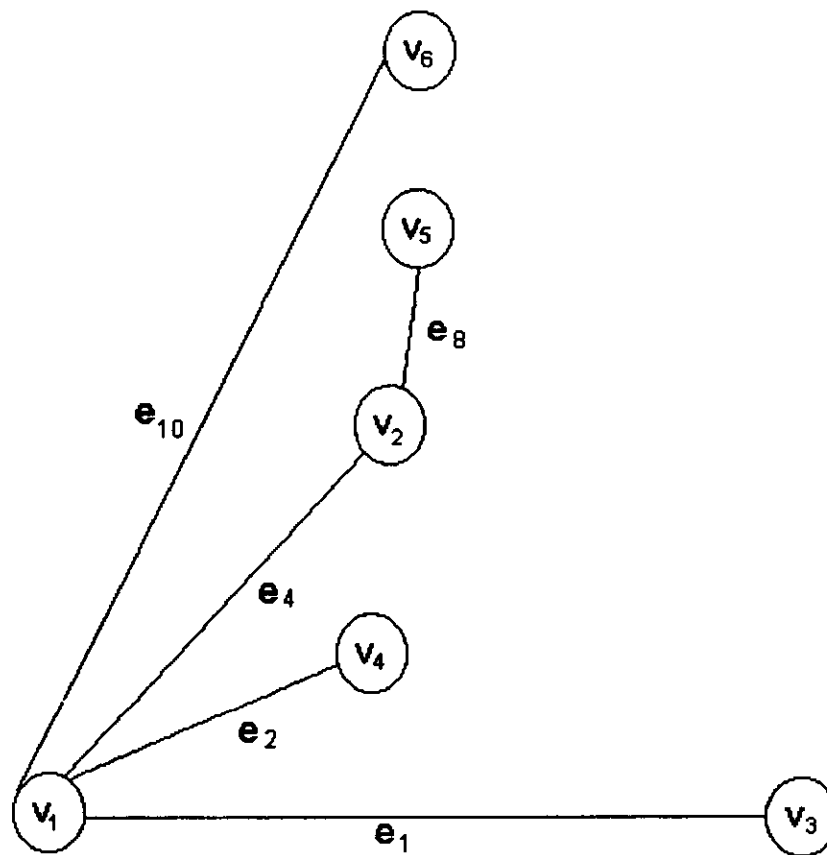


Figura 5.4 3rbol de expansi3n T de la Gr3fica G.

El Arbol de Expansi3n de la Gr3fica Gd.

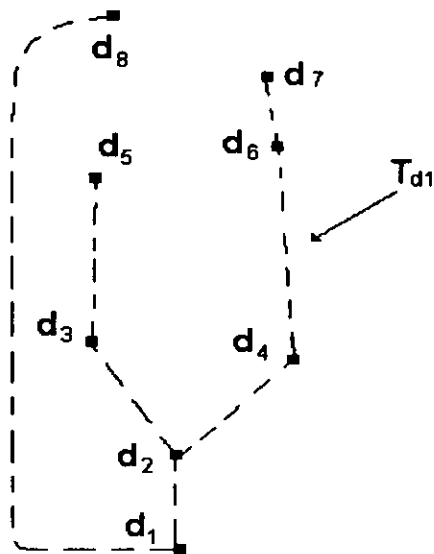


Figura 5.5 Arbol de Expansi3n de la Gr3fica G_d .

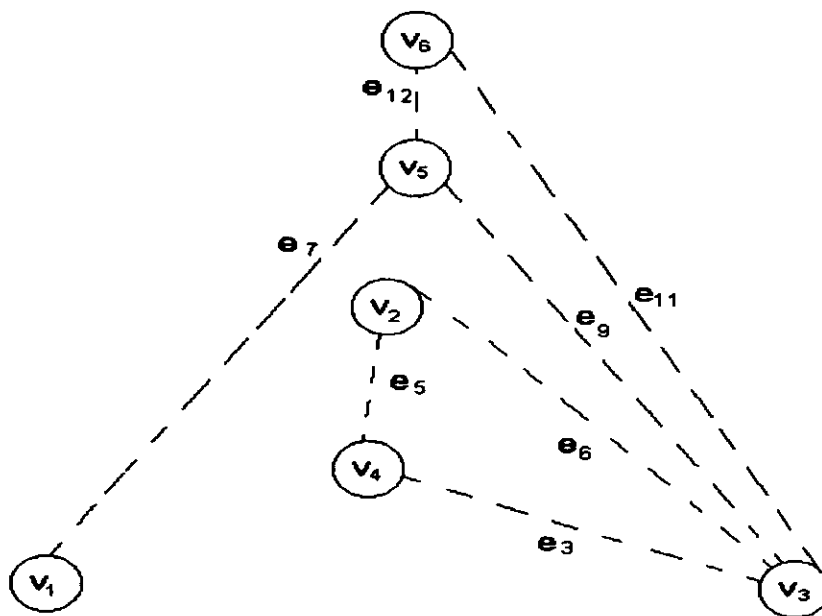


Figura 5.6 Subgr3fica M de cuerdas referentes al 3rbol de Expansi3n T .

En las dos figuras siguientes tenemos dos 3rboles de expansi3n posibles de la gr3fica G_d . T_{d1} y T_{d2} que cortan la Gr3fica G .

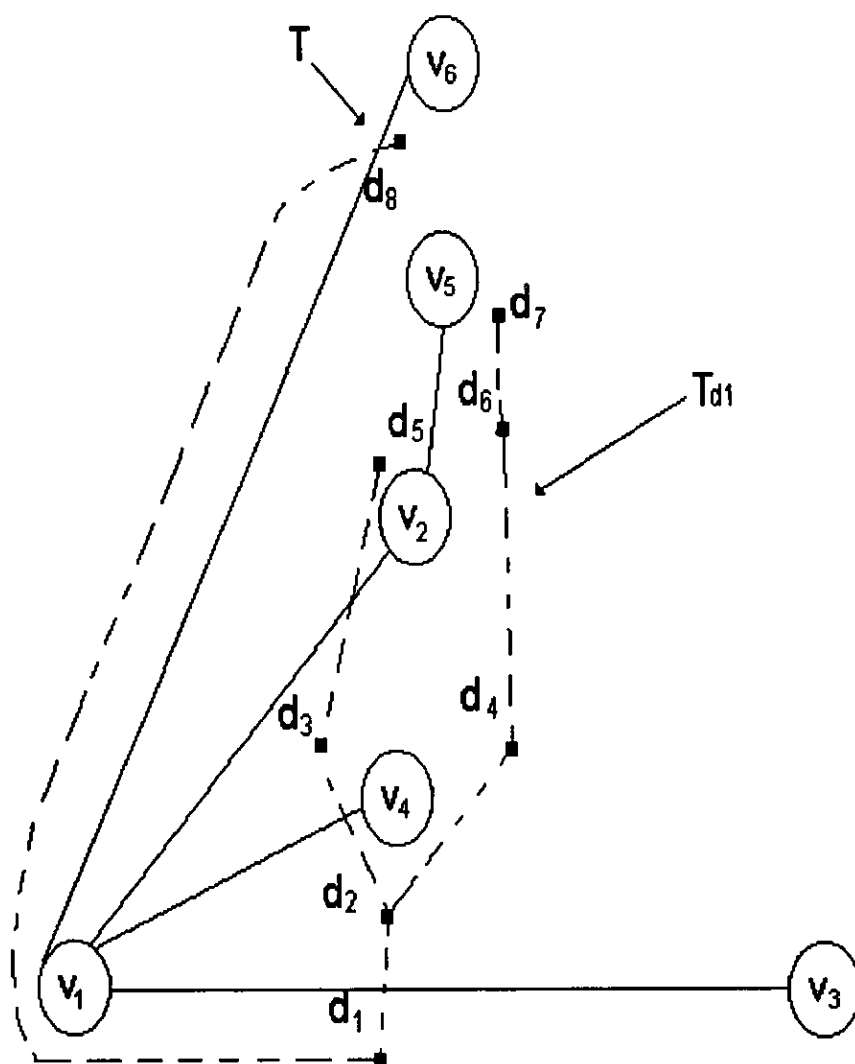


Figura 5.7 Relación entre el árbol T de la gráfica primal G y el árbol T_{d1} de la gráfica dual G_d .

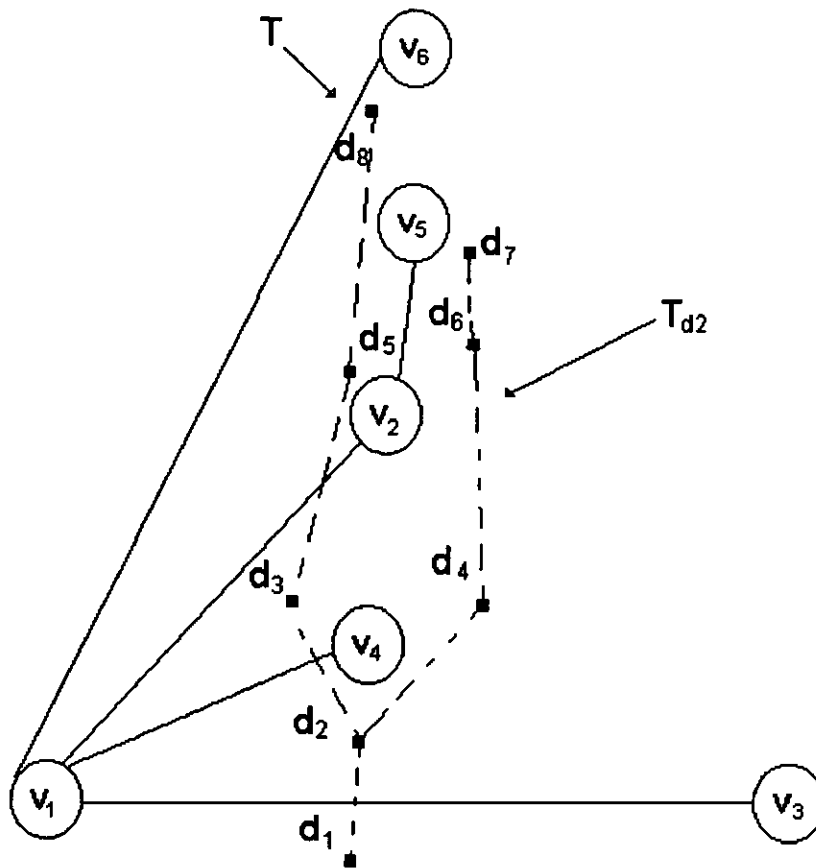


Figura 5.8 Relación entre el árbol T de la gráfica primal G y el árbol T_{d2} de la gráfica dual G_d .

Las ramas del Árbol de expansión de G_d , las cuales cruzan a las ramas del Árbol T de la Gráfica Primal, definen las adyacencias fuertes entre esas áreas.

Al relacionar estos dos árboles T_{d1} y T_{d2} con el árbol T de la gráfica primal G podemos determinar las ramas de T_{d1} y T_{d2} que cortan a las ramas de T , lo cual nos indica las relaciones originales en G que se deberán reforzar.

Para el árbol T_{d1} :

Las relaciones posibles a reforzar son:

$e_1 = (v_1, v_3)$: La docencia ante la formación y capacitación para y en el trabajo.

$e_2 = (v_1, v_4)$: Reestructuración de la normatividad congruente a la flexibilidad de formación y capacitación.

$e_4 = (v_1, v_2)$: Pertinencia del currículum ante las necesidades del sector productivo.

$e_{10} = (v_1, v_6)$: Incorporación inmediata de los egresados al sector productivo (participando en la transformación de la sociedad).

Para el árbol T_{d2} :

Las relaciones posibles a reforzar son:

$e_1 = (v_1, v_3)$: La docencia ante la formación y la capacitación para y en el trabajo.

$e_2 = (v_1, v_4)$: Reestructuración de la normatividad congruente a la flexibilidad de formación y capacitación .

$e_4 = (v_1, v_2)$: Pertinencia del currículum ante las necesidades del sector productivo.

El árbol T_{d2} , es un árbol que esta más centrado con respecto a T_{d1} pero al determinar los cruces con las ramas de T resulta evidente que T_{d1} marca relaciones más relevantes a reforzar sobre todo la relación e_{10} que es una relación entre el punto inicial v_1 y el final v_6 , es decir , la razón de nuestra premisa fundamental que es la conjunción estratégica de la **Educación-Sector Productivo-Sociedad** y el “ producto terminal “, profesionales preparados en áreas de competencias.

Dentro de una estructura lógica, en base a un pensamiento vertical, éste árbol podría ser el más adecuado, pero no tenemos una relación directa con el principio fundamental del MEBC v_1 , con v_3 , v_5 y v_6 .

Partiendo de las relaciones de e_1 : La docencia ante la formación y capacitación para y en el trabajo.

v_1 : Trinomio, Educación-Sector Productivo-Sociedad.

v_3 : Docencia.

e_2 : Reestructuración de la normatividad congruente a la flexibilidad de formación y capacitación.

v_1 : Trinomio, Educación-Sector Productivo-Sociedad.

v_4 : Cultura Organizacional Académica.

Se debe Inferir esta relación e_3 partiendo de las relaciones e_1 y e_2 .

a) Para lograr el área de síntesis d_2 (cumplir con la función social y educativa de la institución (escuela) ante la necesidades del desarrollo del país), las autoridades educativas deben tener conciencia de que el MEBC no cumplirá cabalmente su función al menos que se tengan condiciones interiores, al centro educativo, de trabajo para el docente que le permitan ser ayesado en labores de taller o campo, técnicas pedagógicas así como en áreas disciplinarias , por lo tanto, es necesario que al profesor se le reconozca esta formación.

2) Para la cuerda e_5

Circuito que cierra { e_2 , e_4 , e_5 }

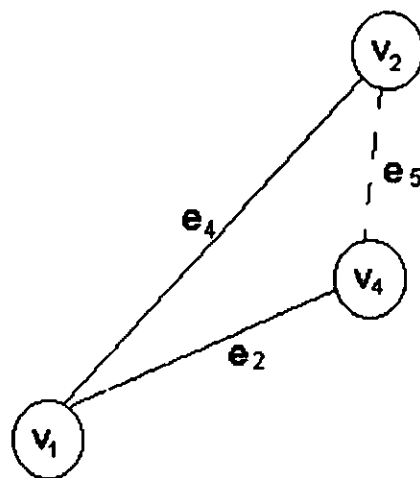


Figura 5.10 Circuito { e_2 , e_4 , e_5 }

Área de síntesis $d_3 \rightarrow \{ v_1, v_2, v_4 \}$

ó

(e_2, e_4, e_5)

$e_5: (v_2, v_4)$

interrelaciona a:

v_2 : Curriculum.

v_4 : Cultura organizacional académica.

Partiendo de las relaciones de:

e_2 : Reestructuración de la normatividad congruente a la flexibilidad de formación y capacitación.

v_1 : Trinomio Educación-Sector Productivo-Sociedad.

v_4 : Cultura Organizacional Académica.

e_4 : Pertinencia del currículum ante las necesidades del Sector Productivo.

v_1 : Trinomio Educación-Sector Productivo-Sociedad.

v_2 : Currículum.

Se infiere: Para lograr esta síntesis d_3 de conocimiento se debe dar la relación de v_2 con v_4 que la podemos interpretar como:

3) Para la cuerda $e_7: (v_2, v_4)$

Circuito que encierra { e_4, e_7, e_8 }

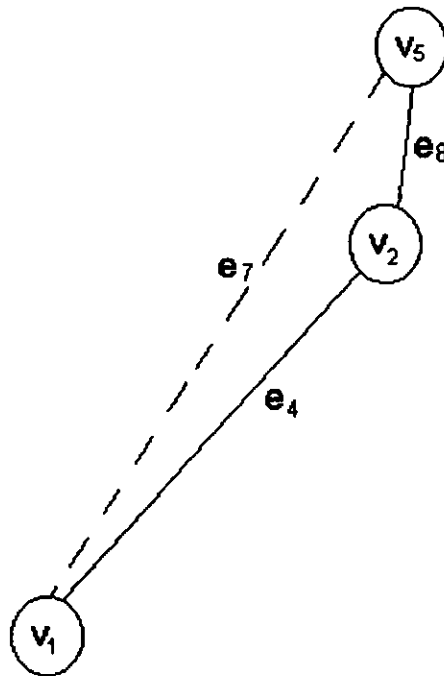


Figura 5.11 Circuito { e4 , e7, e8 }

Área de síntesis $d_5 \{ v_1, v_2, v_6 \}$
 ó
 (e_4, e_7, e_8)

$e_7: (v_1, v_5)$

interrelaciona a:

v_1 : Trinomio Educación - Sector productivo - Sociedad.

v_5 : Salidas laterales.

Partiendo de las relaciones de:

e_4 : Pertinencia del currículum ante las necesidades del Sector Productivo.

v_1 : Trinomio Educación - Sector productivo - Sociedad.

v_3 : Docencia.

$e_8 : (v_2, v_5)$

v_2 : Currículum.

v_5 : Salidas laterales.

Se infiere: Para lograr esta síntesis ds el alumno que egresa en opción terminal debe tener una visión clara de la demanda del sector productivo con respecto a su especialidad la cual él podrá contribuir a satisfacer esta necesidad, apoyado por la formación inducida por el trinomio:

Educación-Sector Productivo-Sociedad.

Circuitos constructivamente inferidos por áreas de síntesis y las relaciones implícitas.

a) Circuito constructivamente inferido por áreas de síntesis y relaciones explícitas.

{ e_3 , e_5 , e_6 }

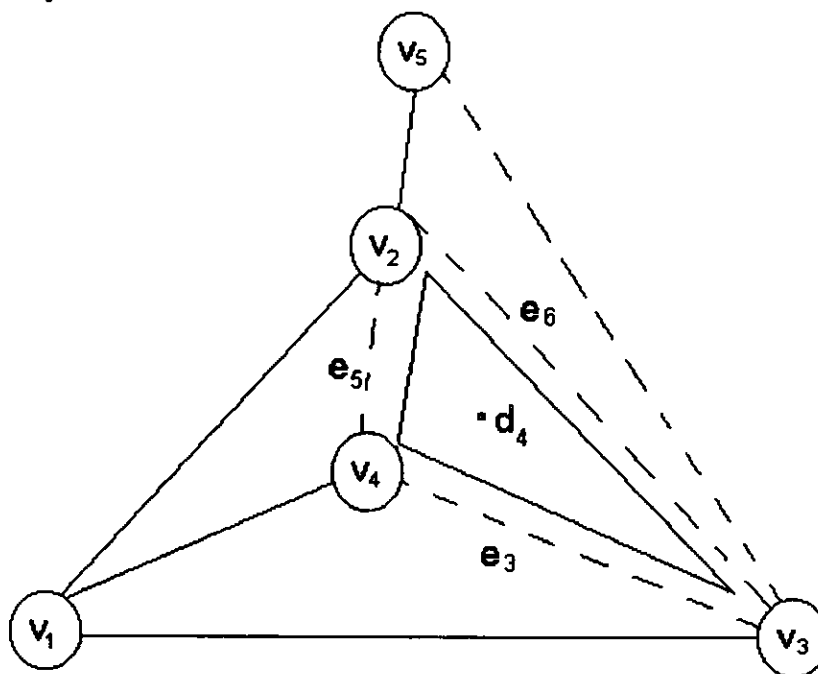


Figura 5.12 Circuito inferido (e_3 , e_5 , e_6)

Área de síntesis inferida:

d_4 : La Cultura Organizacional Académica permite que la docencia participe en el diseño curricular acorde a la demanda del sector productivo.

■ Áreas de síntesis que infieren al circuito:

d3: La Cultura Organizacional Académica permite la pertinencia de los contenidos ofertados en sus diferentes modalidades que satisfacen las necesidades de desarrollo del país.

d6: Salidas laterales demandadas por el sector productivo que se dan gracias a la flexibilidad del curriculum y a la actualización permanente de los docentes.

b) Circuito inferido

{ e3 , e5, e6 }

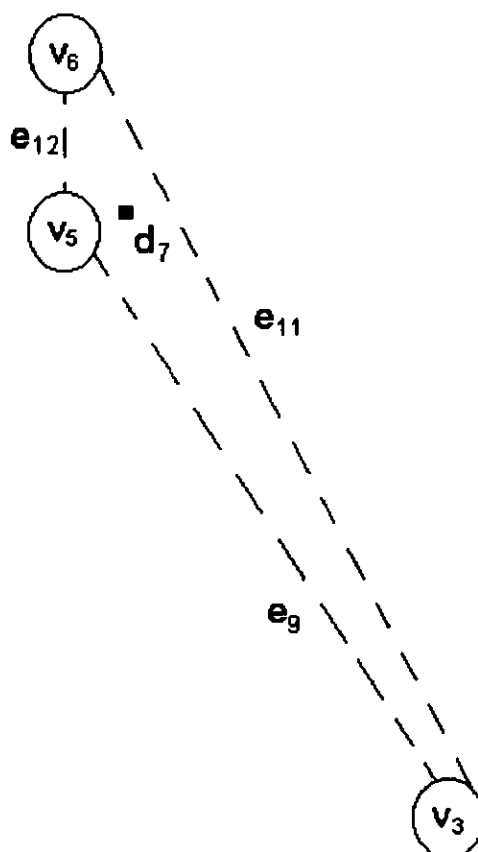


Figura 5.13 Circuito inferido (e3, e5, e6)

Área de síntesis inferida:

d7: Amplia gama de participación del docente en las diferentes modalidades educativas (salidas laterales y licenciatura), exigen del docente una

formación y actualización muy completa y estar incorporado al sector productivo.

■ Áreas de síntesis que infieren al circuito:

d6: Salidas laterales demandadas por el sector productivo que se dan gracias a la flexibilidad del currículum y a la actualización permanente de los docentes.

d8: Todas las modalidades de formación y capacitación corresponden a las necesidades del sector productivo y la sociedad.

Matriz de Circuitos Fundamentales (Programa : C / B / U)

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12
C1	1	1	1	0	0	0	0	0	0	0	0	0
C2	0	1	0	1	1	0	0	0	0	0	0	0
C3	1	0	0	1	0	1	0	0	0	0	0	0
C4	0	0	0	1	0	0	1	1	0	0	0	0
C5	1	0	0	1	0	0	0	1	1	0	0	0
C6	1	0	0	0	0	0	0	0	0	1	1	0
C7	0	0	0	1	0	0	0	1	0	1	0	1

Los circuitos fundamentales son:

C1 = { e1 , e2, e3 }

C2 = { e2 , e4, e5 }

C3 = { e1 , e4, e6 }

C4 = { e4 , e7, e8 }

C5 = { e1 , e4, e8 , e9 }

C6 = { e1 , e10, e11 }

C7 = { e4 , e8, e10 , e12 }

Modificando el orden de las columnas en la matriz C poniendo primero las cuerdas del árbol T.

$$C = \begin{matrix} & \begin{matrix} e_3 & e_5 & e_6 & e_7 & e_9 & e_{11} & e_{12} & e_1 & e_2 & e_4 & e_8 & e_{10} \end{matrix} \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

$$C = \begin{bmatrix} & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \end{bmatrix} I \quad \begin{bmatrix} & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & & & & & \end{bmatrix}$$

- El árbol T de Expansión se expresa como:

$$T = (1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0)$$

- La subgráfica M de cuerdas esta dada por:

$$M = G - T$$

$$M = \{0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1\}$$

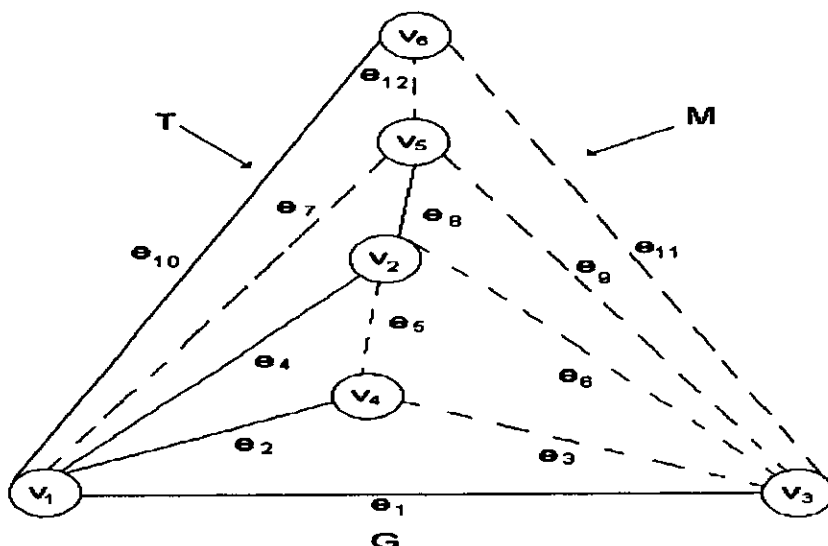


Figura 5.14 Gráficas G con subgráfica T (Árbol de Expansión) y M (formada por cuerdas de T).

Matriz de cortes fundamentales.

Los cortes fundamentales son aquellos que cortan las cuerdas del árbol T.

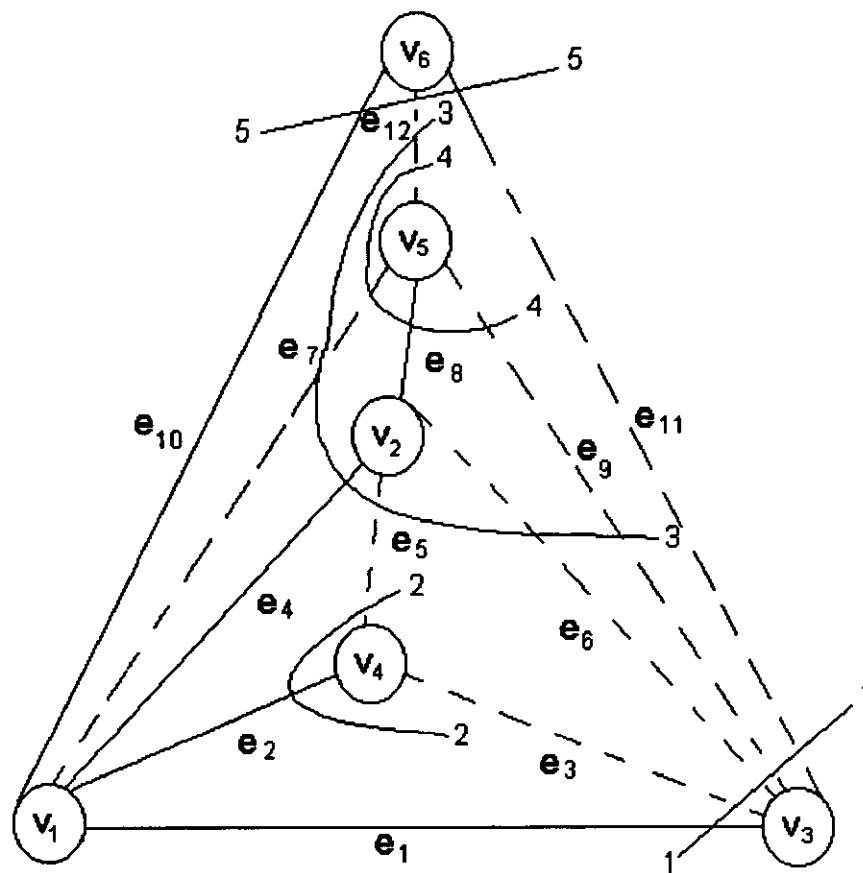


Figura 5.15 Cortes fundamentales

De la matriz de circuitos fundamentales C.

$$C = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_4 & e_8 & e_{10} \end{matrix} \\ \begin{matrix} \boxed{1} \\ C \end{matrix} = & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

$$\begin{vmatrix} C^- \\ I \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$K = \begin{vmatrix} C^- \\ I \end{vmatrix}^T$$

$$K = \begin{matrix} & \begin{matrix} e_3 & e_5 & e_6 & e_7 & e_9 & e_{11} & e_{12} & e_1 & e_2 & e_4 & e_8 & e_{10} \end{matrix} \\ \begin{matrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \end{matrix} & \begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \end{matrix}$$

Los cinco cortes básicos son:

$$k_1 = \{ e_3, e_6, e_9, e_{11}, e_1 \}$$

$$k_2 = \{ e_3, e_5, e_2 \}$$

$$k_3 = \{ e_5, e_6, e_7, e_9, e_{12}, e_4 \}$$

$$k_4 = \{ e_7, e_9, e_{12}, e_8 \}$$

$$k_5 = \{ e_{11}, e_{12}, e_{10} \}$$

Dado que los circuitos y cortes fundamentales forman bases en los subespacios vectoriales C_G y K_G del espacio vectorial E_G que

define a la gráfica G , cualquier circuito C_x o corte K_x se puede expresar como:

$$C_x = a_1c_1 + a_2c_2 + a_3c_3 + a_4c_4 + a_5c_5 + a_6c_6 + a_7c_7$$

$$K_x = a_1k_1 + a_2k_2 + a_3k_3 + a_4k_4 + a_5k_5$$

Donde $a_1, a_2, \dots, a_7 \in \{0, 1\}$

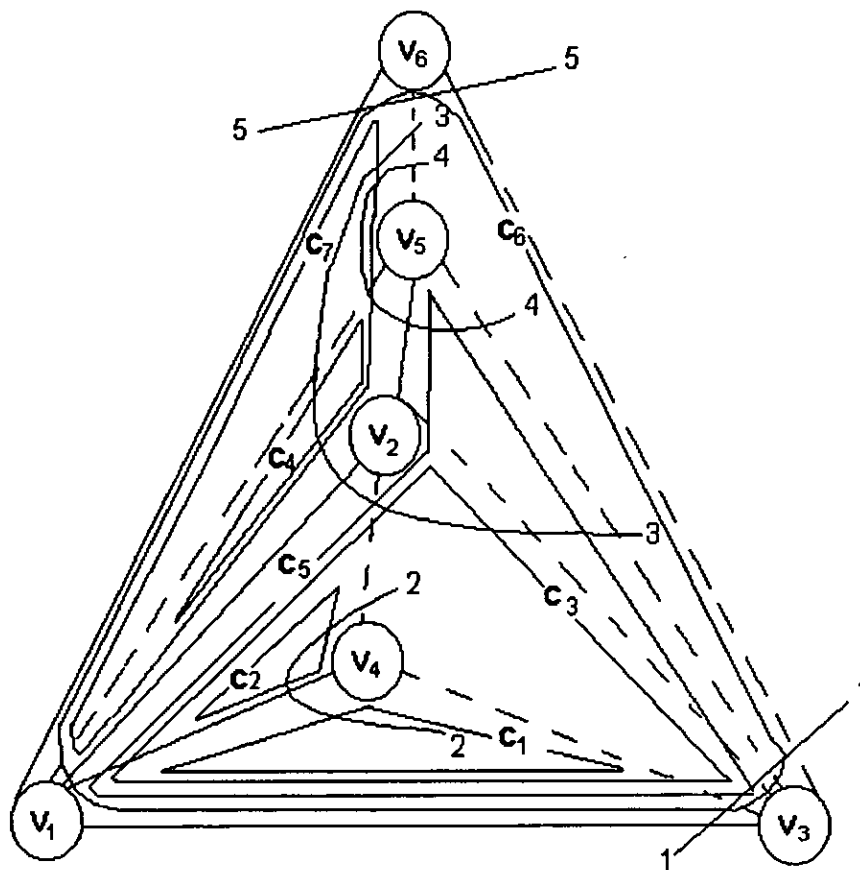


Figura 5.16 Circuitos y cortes fundamentales

En base a estas expresiones generales de C_x y K_x , podemos determinar todos los circuitos y cortes, pero no todos son de interés, es decir, no todos tienen una interpretación práctica o relevante en el contexto del problema. A continuación se determinan circuitos inducidos por los circuitos fundamentales donde su relevancia resulta evidente y además son inferidos por conceptos implícitos.

a) El circuito C_a que se infiere de los conceptos implícitos C_1 , C_2 , C_3 ,

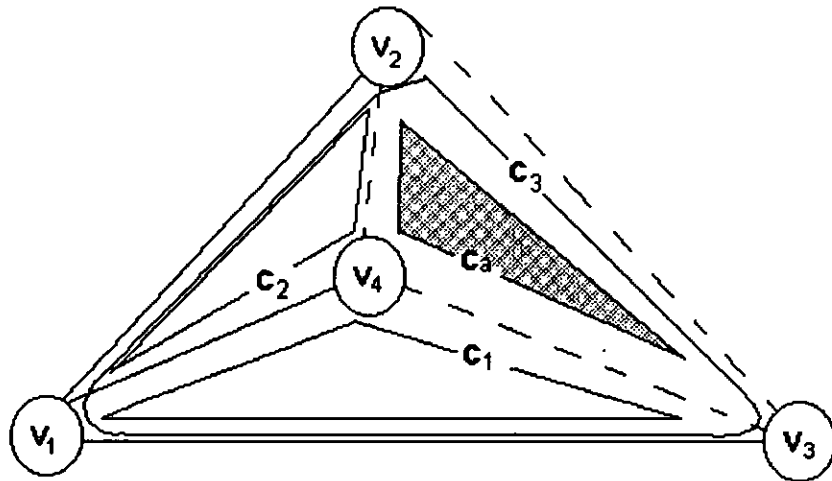


Figura 5.17 Subgráfica del circuito C_a expresado como una combinación lineal de los circuitos básicos C_1 , C_2 , C_3 .

$$C_a = C_1 + C_2 + C_3$$

$$C_a = (e_1, e_2, e_3) + (e_2, e_4, e_5) + (e_1, e_4, e_6)$$

$$C_a = 1 \times C_1 + 1 \times C_2 + 1 \times C_3 + 0 \times C_4 + 0 \times C_5 + 0 \times C_6 + 0 \times C_7$$

$$C_a = C_1 + C_2 + C_3$$

$$C_a = (1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0) +$$

$$(0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0) +$$

$$(1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0)$$

$$C_a = (0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0)$$

$$C_a = (e_3, e_5, e_6)$$

Comentario al circuito C_a :

C_a: La Cultura Organizacional Académica que implica la mística del docente que le permite la participación en el diseño curricular académico acorde a las necesidades del Sector Productivo. Está determinado por las condiciones de trabajo del docente con una formación consistente y un reconocimiento oficial de los estudios realizados por los estudiantes egresados.

b) El circuito C_b que se infiere de los conceptos implícitos C₅ , C₆ , C₇.

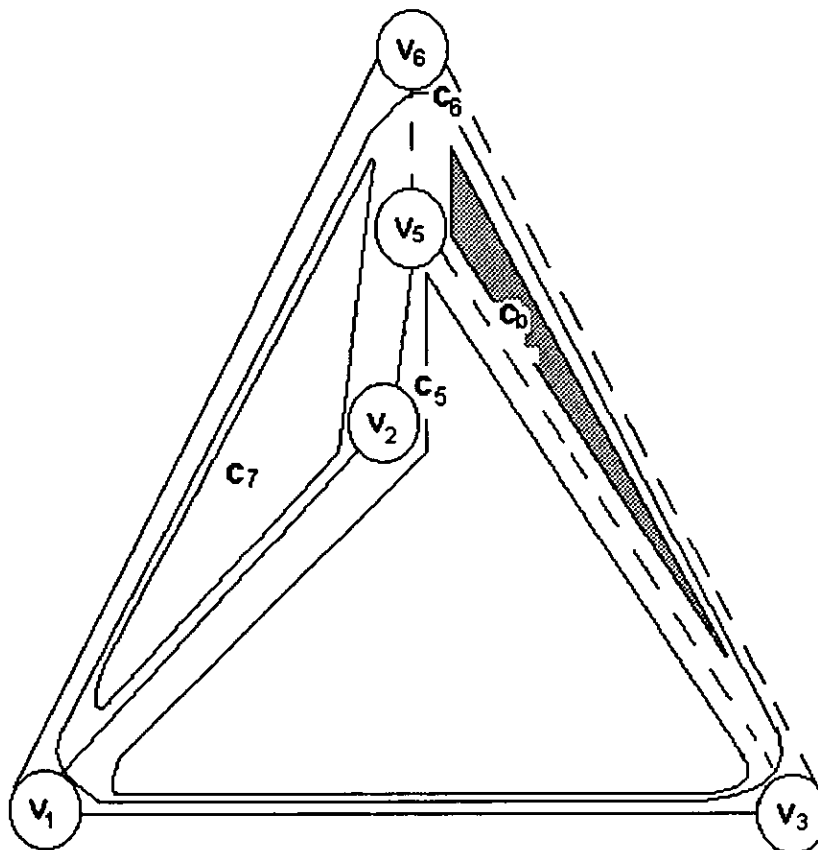


Figura 5.18 Circuito C_b expresado como una combinación lineal de los circuitos C₅ , C₆ , C₇.

$$C_b = C_5 + C_6 + C_7$$

$$C_b = (e_1, e_4, e_8, e_9) + (e_1, e_{10}, e_{11}) + (e_4, e_8, e_{10}, e_{12})$$

$$C_b = 0 \times C_1 + 0 \times C_2 + 0 \times C_3 + 0 \times C_4 + 1 \times C_5 + 1 \times C_6 + 1 \times C_7$$

$$C_b = C_5 + C_6 + C_7$$

$$C_b = (1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0) +$$

$$(1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0) +$$

$$(0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1)$$

$$C_b = (0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1)$$

$$C_b = (e_9, e_{11}, e_{12})$$

Comentario al circuito C_b :

C_b : Para lograr una docencia de excelencia es necesario la actualización docente y será urgente para responder a la oferta educativa a nivel licenciatura solo así se podrá satisfacer la demanda del sector productivo, pero tenemos un elemento adicional que lo proporciona la estructura del MEBC que es el tránsito libre de las Salidas terminales a niveles de licenciatura lo cual proporciona una mayor consistencia de las competencias adquiridas por los estudiantes y una mayor versatilidad del docente.

c) Circuito C_c

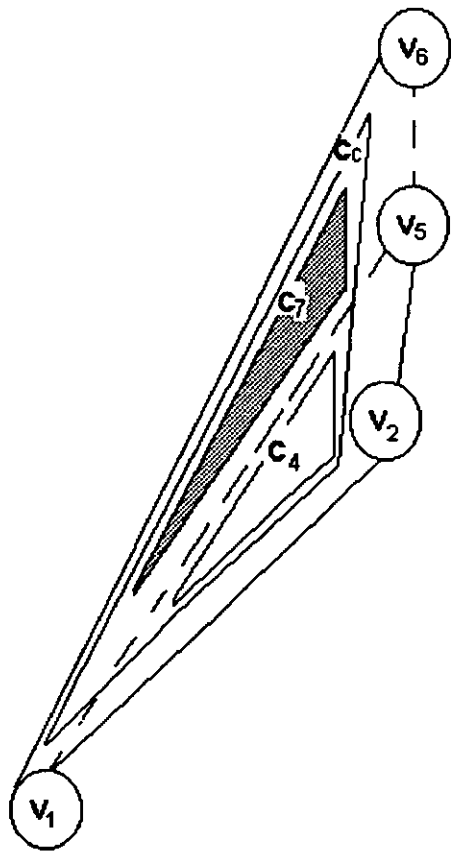


Figura 5.19 Subgráfica del circuito C_c expresado como una combinación lineal de los circuitos básicos C_4 y C_7 .

$$C_c = C_4 + C_7$$

$$C_c = (e_4, e_7, e_8) + (e_4, e_8, e_{10}, e_{12})$$

$$C_c = 0 \times C_1 + 0 \times C_2 + 0 \times C_3 + 1 \times C_4 + 0 \times C_5 + 0 \times C_6 + 1 \times C_7$$

$$C_c = C_4 + C_7$$

$$C_c = (0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0) +$$

$$(0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1)$$

$$C_c = \underline{(0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1)}$$

$$C_c = (e_7, e_{10}, e_{12})$$

Comentario al circuito C_c :

C_c : Con la incorporación en corto plazo del estudiante o egresado, de cualquiera de las modalidades de formación que ofrece el MEBC, así como la posibilidad del libre tránsito entre estas modalidades del MEBC garantiza que se satisficiera la demanda del Sector Productivo de técnicos y profesionales.

d) Circuito C_d .

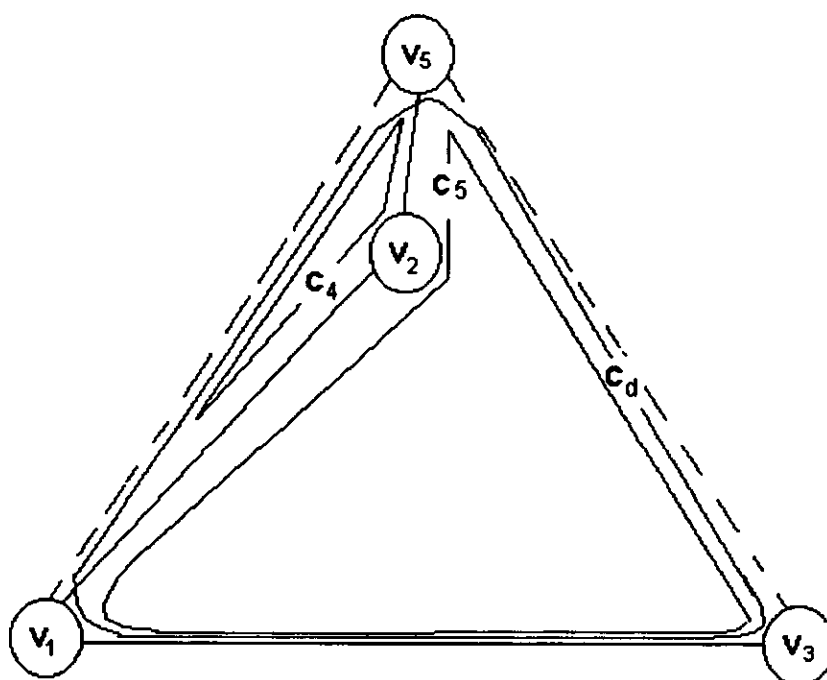


Figura 5.20 Subgráfica del circuito C_d expresado como una combinación lineal de los circuitos básicos C_4 y C_5 .

$$C_d = C_4 + C_5$$

$$C_d = (e_4, e_7, e_8) + (e_1, e_4, e_8, e_9)$$

$$C_d = 0 \times C_1 + 0 \times C_2 + 0 \times C_3 + 1 \times C_4 + 1 \times C_5 + 0 \times C_6 + 0 \times C_7$$

$$C_d = C_4 + C_5$$

$$C_d = (0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0) +$$

(1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0)

$$C_d = \overline{(1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0)}$$

$$C_d = (e_1, e_7, e_9)$$

Comentario al circuito C_d :

C_d : La modalidad de salidas laterales del MEBC permitirá que se satisfaga la demanda del Sector Productivo de una pertinente gracias a la actualización del docente en el campo laboral.

e) Circuito C_e .

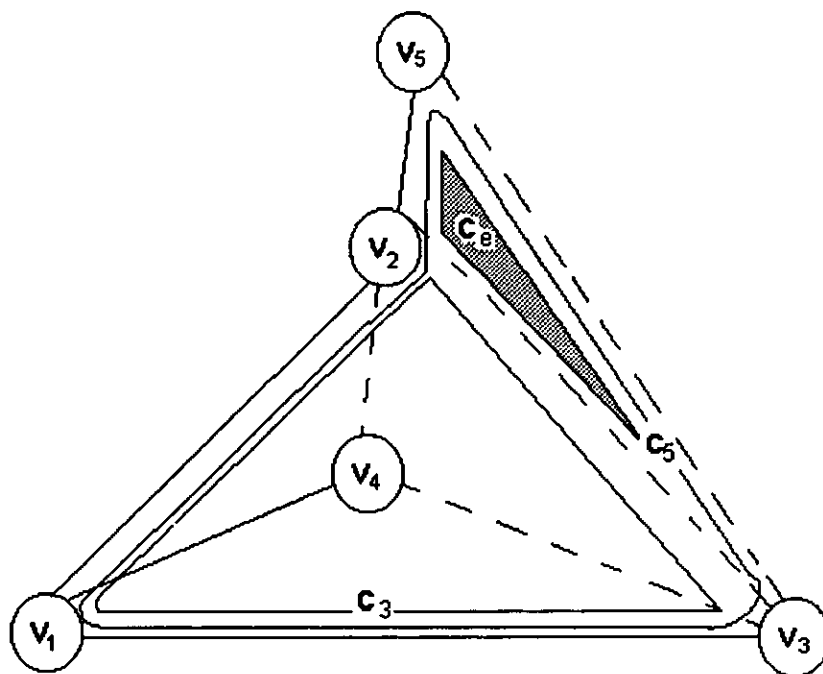


Figura 5.21 Subgráfica del circuito C_e expresado como una combinación lineal de los circuitos básicos C_3 y C_5 .

$$C_e = C_3 + C_5$$

$$C_e = (e_1, e_4, e_6) + (e_1, e_4, e_8, e_9)$$

$$C_e = 0 \times C_1 + 0 \times C_2 + 1 \times C_3 + 0 \times C_4 + 1 \times C_5 + 0 \times C_6 + 0 \times C_7.$$

$$C_e = (1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0) +$$

$$(1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0)$$

$$C_e = \frac{(0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0)}{}$$

$$C_e = (e_6, e_8, e_9)$$

Comentario al circuito C_e :

C_e : Técnicos demandados por el sector productivo de las salidas laterales, que se dan gracias a la flexibilidad del currículum y a la actualización permanente del docente.

CAPÍTULO 6

6. CONCLUSIONES.

Los elementos de la gráfica que determinan los vértices (v_1 v_6) son definidos básicamente en los ejes estructurales del MEBC. Obviamente la relación entre ellos debe ser muy fuerte pues los términos definidos a estas relaciones nos denotan las acciones o efectos en el contexto general. Estas relaciones son lógicas y evidentes pero podrían ser cambiadas según la evaluación del modelo (se espera tener una evaluación por lo menos en los próximos cinco años) y las opiniones del personal operativo y sectores involucrados en el sector educativo, sector social y sector productivo. Podemos definir términos claves, con respecto a las relaciones, para poder referirnos en el resultado del análisis:

- e1: Docencia para el trabajo.
- e2: Normatividad congruente.
- e3: Congruencia laboral.
- e4: Pertinencia ante la producción.
- e5: Validez oficial.
- e6: Actualización docente (pedagógica y disciplinaria).
- e7: Incorporación inmediata de salidas laterales.
- e8: Currículum adecuado.
- e9: Instructor. Docente-Instructor.
- e10: Incorporación inmediata de profesionales.
- e11: Docente-Instructor para Licenciatura.
- e12: Tránsito libre.

Áreas de síntesis.

De las regiones conformadas en la gráfica tenemos:

Región d1.

Partimos de que el contexto general es una situación de crisis que solo se puede remontar aumentando por medio de una educación congruente con las necesidades de la producción.

Región d2.

En ésta área de síntesis concluimos que para tener una docencia que induzca directamente al trabajo es determinante tener una **normatividad congruente** en la institución educativa con respecto del reconocimiento de las competencias adquiridas y a su vez una **congruencia laboral** en el sentido de que el docente, no será un profesor tradicional ya que debe estar capacitado en el ámbito pedagógico y en su disciplina correspondiente a nivel teórico - práctico.

Región d3.

La pertenencia ante el sector productivo de la educación está determinada por las condiciones laborales de los docentes (Normatividad congruente) y el reconocimiento oficial de las competencias adquiridas por los egresados, en el campo laboral o en la escuela.

Región d4.

La actualización pedagógica y disciplinaria del docente implica una nueva reglamentación laboral, ya que esta por lo menos la capacitación disciplinaria por lo menos en buena parte deberá ser en los centros de producción y esta actividad debe ser reconocida oficialmente, es decir, debe existir una congruencia en sus condiciones laborales con las actividades desarrolladas para su actualización y solo así los conocimientos impartidos por el docente podrán tener validez oficial.

Región d5.

Con la pertenencia ante el sector productivo, de los conocimientos y competencias contenidas en un currículo adecuado, se asegura la incorporación inmediata de los técnicos egresados, a la producción.

Región d6.

El MEBC exige que el docente haga una actualización permanente en dos áreas básicamente; la pedagógica y la disciplinaria (su especialidad en el desempeño profesional) si su quehacer académico está basado en un currículo adecuado se podrá tener una plantilla docente de excelencia, también exigencia del MEBC.

Región d7.

La amplia gama de posibilidades que ofrece el MEBC, para el desempeño del docente, exige que este sea un docente de excelencia, considerando que se tiene tránsito libre entre los diferentes niveles educacionales, también obliga al docente a una preparación diferente a la tradicional.

Región d8.

Los egresados de la escuela con un MEBC en cualquier nivel deben poder incorporarse al sector productivo de inmediato. Esto implica que el docente deberá estar capacitado para atender cualquier nivel educativo ofertado por la escuela.

Resumen de las relaciones relevantes, de Circuitos Básicos y Circuitos Inferidos, así como Áreas de Síntesis.

1. Ramas del Árbol de Expansión.

$$1.1 \cdot e_1 = (v_1 , v_3)$$

1. Término: La docencia ante la formación y capacitación para el trabajo.

$$1.2 \cdot e_2 = (v_1 , v_4)$$

2. Término: Reestructuración de la normatividad congruente a la flexibilidad de formación y capacitación.

$$1.3 \cdot e_4 = (v_1 , v_2)$$

3. Término: Pertinencia del curriculum ante las necesidades del Sector Productivo.

$$1.4 \ e_8 = (v_2 , v_5)$$

4. Término: Currículum flexible y polivalente.

$$1.5 \ e_{10} = (v_1 , v_6)$$

5. Término: Incorporación inmediata de los egresados al Sector Productivo, participando en la transformación de la Sociedad.

2 Cuerdas del Árbol de Expansión.

$$2.1 \ e_3 = (v_1 , v_4)$$

6. Término: Congruencia de las condiciones interiores de trabajo para la versatilidad de la oferta educativa que brinda el MEBC.

$$2.2 \ e_5 = (v_2 , v_4)$$

7. Término: Válidez oficial de los egresados de las diferentes modalidades educativas.

$$2.3 \ e_6 = (v_2 , v_3)$$

8. Término: Formación (pedagógica) y actualización (disciplinaria) del docente para atender con calidad a lo establecido en el currículum.

$$2.4 \ e_7 = (v_1 , v_5)$$

9. Término: Incorporación a corto plazo al Sector Productivo.

$$2.5 \ e_9 = (v_3 , v_5)$$

10. Término: La docencia altamente capacitada y la vanguardia de la tecnología que permita al egresado de cualquier salida lateral incorporarse al Sector Productivo.

$$2.6 \ e_{11} = (v_3, v_6)$$

11. Término: La docencia frente a la permanente actualización y vigencia de la oferta educativa a nivel licenciatura que responda a la demanda del Sector Productivo.

$$2.7 \ e_{12} = (v_5, v_6)$$

12. Término: Posibilidad de tránsito de las salidas laterales al nivel licenciatura.

3 Áreas de Síntesis.

$$3.1 \ d_1 \rightarrow$$

13. Término: Efecto, en un contexto de crisis, de profesionales preparados a nivel licenciatura o técnico medio capacitado para adquirir competencias necesarias para la producción y desarrollo de la sociedad .

$$3.2 \ d_2 \rightarrow \begin{matrix} (v_1, v_3, v_4) \\ \text{ó} \\ (e_1, e_2, e_3) \end{matrix}$$

14. Término: Cumplir con la función social y educativa de la institución ante las necesidades económicas y sociales del país.

$$3.3 \ d_3 \rightarrow \begin{matrix} (v_1, v_2, v_4) \\ \text{ó} \\ (e_2, e_4, e_5) \end{matrix}$$

15. Término: La Cultura Organizacional Académica permite la pertinencia de los contenidos ofertados en sus diferentes modalidades que satisfacen las necesidades de desarrollo del país.

$$3.4 \ d_4 \rightarrow \begin{matrix} (v_2, v_3, v_4) \\ \text{ó} \\ (e_3, e_5, e_6) \end{matrix}$$

16. Término: La Cultura Organizacional Académica permite que la docencia participe en el diseño curricular acorde a la demanda del sector productivo.

$$\begin{array}{c} 3.5 \ d_5 \rightarrow \quad (v_1, v_2, v_5) \\ \qquad \qquad \qquad \quad \text{ó} \\ \qquad \qquad \qquad \quad (e_4, e_7, e_8) \end{array}$$

17. Término: Flexibilidad del currículum a través de salidas laterales que permiten incorporar a los egresados en forma inmediata al sector productivo.

$$\begin{array}{c} 3.6 \ d_6 \rightarrow \quad (v_3, v_2, v_5) \\ \qquad \qquad \qquad \quad \text{ó} \\ \qquad \qquad \qquad \quad (e_6, e_9, e_4) \end{array}$$

18. Término: Salidas laterales demandadas por el sector productivo que se dan gracias a la flexibilidad del currículum y a la actualización permanente de los docentes.

$$\begin{array}{c} 3.7 \ d_7 \rightarrow \quad (v_3, v_5, v_6) \\ \qquad \qquad \qquad \quad \text{ó} \\ \qquad \qquad \qquad \quad (e_9, e_{11}, e_{12}) \end{array}$$

19. Término: Amplia gama de participación del docente en las diferentes modalidades educativas (salidas laterales y licenciatura), exigen del docente una formación y actualización muy completa y estar incorporado al sector productivo.

$$\begin{array}{c} 3.8 \ d_8 \rightarrow \quad (v_1, v_5, v_6) \\ \qquad \qquad \qquad \quad \text{ó} \\ \qquad \qquad \qquad \quad (e_7, e_{10}, e_{11}) \end{array}$$

20. Término: Todas las modalidades de formación y capacitación corresponden a las necesidades del sector productivo y la sociedad.

4 Circuitos fundamentales.

$$\begin{array}{c} 4.1 \ C_1 = \quad (v_1, v_3, v_4) \\ \qquad \qquad \qquad \quad \text{ó} \end{array}$$

(e₁ , e₂ , e₃)

21.Término: Congruencia de las condiciones internas de trabajo con la versatilidad de la oferta educativa que brinda el MEBC.

4.2 C₂ = (v₁, v₂, v₄)
ó
(e₂ , e₃ , e₅)

22. Término: Válidez oficial de los estudios realizados por los egresados de las diferentes modalidades educativas.

4.3 C₃ = (v₁ , v₂, v₃)
ó
(e₁ , e₄ , e₆)

23.Término: Formación (pedagógica) y actualización disciplinaria permanente del docente para atender con calidad a lo establecido en el currículum.

4.4 C₄ = (v₁, v₂, v₅)
ó
(e₄, e₇, e₈)

24.Término: Incorporación del estudiante o egresado en corto plazo al Sector Productivo.

4.5 C₅ = (v₁, v₂, v₅, v₃)
ó
(e₄, e₈, e₁₂)

25.Término: Docencia altamente capacitada y a la vanguardia de la tecnología que permita al egresado de cualquier salida lateral incorporarse al Sector Productivo.

4.6 C₃ = (v₃, v₅, v₆)
ó
(e₁, e₁₀, e₁₁)

26. Término: La docencia frente a la permanente actualización y urgencia de la oferta educativa a un nivel licenciatura a la demanda del Sector Productivo.

$$4.7 \ C7 = (V1, V2, V5, V6) \\ \text{ó} \\ (e4, e8, e12)$$

27.Término: Posibilidad de transitar de las salidas laterales a la terminación de una licenciatura.

5 Circuitos inferidos.

$$5.1 \ C_a = (V2, V3, V4) \\ \text{ó} \\ (e3, e5, e6)$$

28. Término: La Cultura Organizacional Académica que implica la mística del docente que le permite la participación en el diseño curricular académico acorde a las necesidades del Sector Productivo. Está determinado por las condiciones de trabajo del docente con una formación consistente y un reconocimiento oficial de los estudios realizados por los estudiantes egresados.

$$5.2 \ C_b = (V3, V5, V6) \\ \text{ó} \\ (e9, e11, e12)$$

29.Término: Para lograr una docencia de excelencia es necesario la actualización docente y será urgente para responder a la oferta educativa a nivel licenciatura solo así se podrá satisfacer la demanda del sector productivo, pero tenemos un elemento adicional que lo proporciona la estructura del MEBC que es el tránsito libre de las Salidas terminales a niveles de licenciatura lo cual proporciona una mayor consistencia de las competencias adquiridas por los estudiantes y una mayor versatilidad del docente.

$$5.3 \ C_c = (V1, V5, V6) \\ \text{ó}$$

(e7, e10, e12)

30. Término: Con la incorporación en corto plazo del estudiante o egresado, de cualquiera de las modalidades de formación que ofrece el MEBC, así como la posibilidad del libre tránsito entre estas modalidades del MEBC garantiza que se satisficiera la demanda del Sector Productivo de técnicos y profesionales.

5.4 $C_d = (V_1, V_3, V_5)$
 ó
 (e1, e7, e9)

31. Término: La modalidad de salidas laterales del MEBC permitirá que se satisfaga la demanda del Sector Productivo de una pertinente gracias a la actualización del docente en el campo laboral.

5.5 $C_e = (V_3, V_8, V_9)$
 ó
 (e6, e8, e4)

32. Término: Técnicos demandados por el sector productivo de las salidas laterales, que se dan gracias a la flexibilidad del currículum y a la actualización permanente del docente.

Resumén tabular de los circuitos básicos y circuitos inferidos.

Rama en el árbol T	Cuerda del árbol T, relación inferida que encierra el circuito básico	Circuito básico	Inferencia de síntesis en circuitos inferidos
$e_1 = (v_1, v_3)$	$e_3 = (v_3, v_4)$ ¹	$C_1 = \{ e_1, e_2, e_3 \}$	
	$e_6 = (v_2, v_3)$ ²	$C_3 = \{ e_1, e_4, e_6 \}$	$C_a = \{ e_3, e_5, e_6 \}$
	$e_9 = (v_3, v_5)$ ³	$C_5 = \{ e_1, e_4, e_8, e_9 \}$	$C_d = \{ e_1, e_7, e_9 \}$ $C_e = \{ e_6, e_8, e_9 \}$
	$e_{11} = (v_3, v_6)$ ⁴	$C_6 = \{ e_1, e_{10}, e_{11} \}$	$C_b = \{ e_9, e_{11}, e_{12} \}$
$e_2 = (v_1, v_4)$	$e_3 = (v_3, v_4)$ ¹	$C_1 = \{ e_1, e_2, e_3 \}$	
	$e_5 = (v_2, v_4)$ ⁵	$C_2 = \{ e_2, e_4, e_5 \}$	
$e_4 = (v_1, v_2)$	$e_5 = (v_2, v_4)$ ⁵	$C_2 = \{ e_2, e_4, e_5 \}$	
	$e_6 = (v_2, v_3)$ ²	$C_3 = \{ e_1, e_4, e_6 \}$	$C_a = \{ e_3, e_5, e_6 \}$
	$e_7 = (v_1, v_5)$ ⁶	$C_4 = \{ e_4, e_7, e_8 \}$	
	$e_9 = (v_3, v_5)$ ³	$C_5 = \{ e_1, e_4, e_8, e_9 \}$	$C_d = \{ e_1, e_7, e_9 \}$ $C_e = \{ e_6, e_8, e_9 \}$
	$e_{12} = (v_5, v_6)$ ⁷	$C_7 = \{ e_4, e_8, e_{10}, e_{12} \}$	$C_b = \{ e_9, e_{11}, e_{12} \}$ $C_c = \{ e_7, e_{10}, e_{12} \}$
$e_8 = (v_2, v_5)$	$e_9 = (v_3, v_5)$ ³	$C_5 = \{ e_1, e_4, e_8, e_9 \}$	$C_d = \{ e_1, e_7, e_9 \}$ $C_e = \{ e_6, e_8, e_9 \}$
	$e_7 = (v_1, v_5)$ ⁶	$C_4 = \{ e_4, e_7, e_8 \}$	
$e_8 = (v_2, v_5)$	$e_{12} = (v_5, v_6)$ ⁷	$C_7 = \{ e_4, e_8, e_{10}, e_{12} \}$	$C_b = \{ e_9, e_{11}, e_{12} \}$ $C_c = \{ e_7, e_{10}, e_{12} \}$
$e_{10} = (v_1, v_6)$	$e_{12} = (v_5, v_6)$ ⁷	$C_7 = \{ e_4, e_8, e_{10}, e_{12} \}$	$C_b = \{ e_9, e_{11}, e_{12} \}$ $C_c = \{ e_7, e_{10}, e_{12} \}$
	$e_{11} = (v_3, v_6)$ ⁴	$C_6 = \{ e_1, e_{10}, e_{11} \}$	$C_b = \{ e_9, e_{11}, e_{12} \}$

Es evidente que al analizar la gráfica propuesta resulta un número elevado de combinaciones, pero podrán establecerse otras gráficas con diferentes conexiones, aunado a este tenemos también un número elevado de Circuitos Inferidos que resultan de las combinaciones lineales de los correspondientes. Como se menciono anteriormente el análisis del MEBC por medio de la Teoría de Gráficas se puede realizar a cualquier nivel, es decir, a nivel estructural como se propone en este trabajo o incluirse a nivel curricular. Agotar las posibilidades de la Teoría de Gráficas resulta prácticamente imposible; pero si puede acotar el problema ya que muchas relaciones y atributos del modelo detectados en el análisis no resultan relevantes pero esto es tarea de profesionales en el quehacer educativo, por lo cual se propone que los grupos de trabajo, que ya existen en los diferentes centros educativos del IPN; dedicados a implantar el MEBC estén apoyados con la propuesta presentada en este trabajo.

APENDICE A

A. ALGORITMOS PROGRAMADOS EN LENGUAJE C, PARA DETERMINAR:

En esta sección se proporcionan algoritmos programados en el lenguaje de Programación C. Identifican en una gráfica dirigida $G (V, A)$ o no dirigida $G (V, E)$:

- Rutas.
- Rutas más cortas
- Rutas más largas.
- Matriz de Circuitos Básicos.
- Circuito de Euler.
- Circuito de Hamilton.
- Árbol de Expansión.
- Árbol de Expansión más corto.
- Árbol de Expansión más largo.

1) Programa: (P/S/D).

Determina una ruta de un vértice inicial v_p a un vértice final v_q en una gráfica dirigida $G (V, A)$, la cual es descrita por su matriz de Adyacencia S .

2) Programa: (P/B/D).

Dada una Matriz de Incidencias B para la gráfica dirigida $G (V, A)$, obtiene una ruta del vértice v_p al vértice v_q .

3) Programa: (PS-1/L/D) - Dijkstra - .

Determina la ruta más corta del vértice v_p al vértice v_q de una gráfica dirigida $G = (V, A)$ representada por la matriz L de distancias.

4) Programa: (PS-2/L/D).

Determina la ruta más corta del vértice v_p al vértice v_q de una gráfica dirigida $G = (V, A)$ representada por la matriz L de distancias.

5) Programa: (PS-3/L/D) - Floyd - .

En base a las Matrices L de Distancias y S de Adyacencias, obtiene como resultado una Matriz L^+ Modificada de distancias, en la cual cada elemento contiene la distancia más corta entre los vértices i y j . La Matriz R de ruta, da la secuencia de vértices de la ruta con distancias más cortas.

6) Programa: (PH/S/D) - Kaoufmann - .
(Método de la Multiplicación Latina)

Determina una ruta de Hamilton a partir de la Matriz de Adyacencias S de una gráfica dirigida $G = (V, A)$ a las rutas simples de G en el caso que no exista la ruta de Hamilton.

7) Programa: (PL/L/D).

Identifica la ruta más larga a partir de la Matriz de Distancias L de una gráfica dirigida $G (V, A)$.

8) Programa: (C/B/U).

Obtiene una Matriz de Circuitos Básicos C de una gráfica no dirigida $G (V, E)$ a partir de la matriz de incidencias B .

9) Programa: (CE/B/U).

A partir de la Matriz de Incidencias B de una gráfica no dirigida $G (V, E)$ sin rizados, localiza un Circuito de Euler.

10) Programa: (CE/S/U).

A partir de la Matriz de Adyacencias S de una gráfica no dirigida $G (V, E)$ sin rizados determina un Circuito de Euler.

11) Programa: (CE/S/D).

A partir de la Matriz de Adyacencias S de una gráfica dirigida $G (V, A)$ sin rizados localiza un Circuito de Euler.

12) Programa: (T-1/B/U).

Determina un Árbol de Expansión de la Gráfica no dirigida $G = (V, E)$ expresada en términos de la Matriz B de Incidencias.

13) Programa: (T-2/S/U).

A partir de la Matriz de Adyacencias de una gráfica $G (V, E)$ encuentra un Árbol de Expansión en términos de una Matriz B de Incidencias.

14) Programa: (T-3/S/D).

Determina un Árbol de Expansión T para una gráfica dirigida $G (V, A)$ representada por la Matriz de Adyacencias S .

15) Programa: (TSS-1/L/U) - Kruskal - .

Determina el Árbol de Expansión mas corto a partir de la Matriz de Distancias L para una gráfica no dirigida $G = (V, E)$.

16) Programa: (TSS-2/L/U) - Solln - .

Determina el Árbol de Expansión mas corto partiendo de la Matriz de Distancias L para una gráfica no dirigida $G = (V, E)$; se recomienda este algoritmo cuando el número de vértices de la gráfica es grande .


```

/*
*****
Algoritmo ( P/S/D )
Encuentra: Una Ruta.
Dado: Matriz de Adyacencias.
Autor: Ing.TOMAS OLIVER SANCHEZ.
Fecha: 22 - MAR - 1997
*****

*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;
FILE *a2;
FILE *a3;

int lee,m,noesta;
int i,j,ii,k,reg,OK;
char dummy[80];

int matrix[20][20]; /* Matriz de Adyacencias */
int T[20]; /* Vertices checados */
int P[20]; /* Vertices precedentes */
int M[20]; /* Vertices de la ruta 2 */
int O[20]; /* Ruta 2 */
int N[20]; /* Ruta Buscada */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("adjacen.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\\n]\\n",dummy);

        /* Lee el numero de vertices y el vertice inicial */

        fscanf(a1,"%d %d\\n",&m,&T[1]);

        /* Lee los elementos de la Matrix de Adyacencias */

        for(i=1;i<=m;i++)
            for(j=1;j<=m;j++)
            {
                fscanf(a1,"%d",&matrix[i][j]);
            }

        /* Imprime resultados */

        system("cls");
        /* Algoritmo ( P/S/D ) - 2 */

        printf("\\n\\n\\tAlgoritmo para determinar una ruta a partir de la Matriz de Adyacencias\\n\\n");
        printf("\\t\\t\\t Matrix de Adyacencias\\n\\n");
        printf("\\t ");
        for(i=1;i<=m;i++) printf(" V%d ",i);
        printf("\\n\\n");
    }
}

```

```

for(i=1;i<=m;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        printf(" %3d ",matrix[i][j]);
    }
    printf("\n");
}
printf("\n");

```

/ Algoritmo para determinar una ruta a partir de la Matriz de Adyacencias */*

/ Paso 1 Asigna valores iniciales */*

reg = 1;

/ T[reg] = 1; Vertice inicial */*

P[reg] = 0; / Como ningun vertice precede al Vertice Inicial se toma el valor de 0 */*

/ Paso 2 Checa cada uno de los renglones y si el elemento es diferente de cero :*

*Agrega el numero de Columna a T
y el numero de renglon a P */*

```

ii := T[1];
for(i=1;i<=m;i++)
{
    for(j=1;j<=m;j++)
    {
        noesta = 0;
        if(matrix[ii][j]>0) /* si el elemento es diferente de cero */
        {
            /* chequea si ya se tiene el vertice */

            for(k=1;k<=m;k++) if(T[k]==j) noesta = 1;

            /* si el vertice no esta repetido */

            if(noesta==0)
            {
                reg = reg + 1;
                T[reg] = j; /* agrega el numero de columna */
                P[reg] = ii; /* agrega el numero de renglon */
            }
        }
    }
}

```

/ Algoritmo (P/S/D) - 3 */*

```

}
}
ii := ii + 1;
if( i>m) ii = 1;
}

reg := 0;
for(i:=1;i<=m;i++)
{
    OK = 0;
    for(j=1;j<=m;j++)

```

```

    {
        if(T[i]==P[j] && OK==0)
        {
            reg = reg + 1;
            N[reg] = T[i];
            OK = 1;
        }
    }
}
reg = reg + 1;
N[reg] = T[m];

printf("\t\tVertices Checados\n\n");
printf("\tT = {");

for(i=1;i<=m;i++)
{
    if(T[i]==0) continue;
    printf(" V%d ",T[i]);
}

printf("\n\n");

printf("\t\tVertices Precedentes\n\n");
printf("\tP = {");
printf(" 0 ");
for(i=2;i<=m;i++)
{
    if(P[i]==0) continue;
    printf(" V%d ",P[i]);
}
printf("\n\n");

for(i=2;i<=m;i++) T[i] = 0;

reg = 1;
P[reg] = 0; /* Como ningun vertice precede al Vertice Inicial se toma el valor de 0 */

/* Paso 2 Checa cada uno de los renglones y si el elemento
es diferente de cero :
/* Algoritmo ( P/S/D ) - 4 */

    Agrega el numero de Columna a T
    y el numero de renglon a P */

ii = T[1];
OK = 1;

for(i=1;i<=m;i++)
{
    if(OK==0 && T[1]>0) continue;
    for(j=1;j<=m;j++)
    {
        noesta = 0;
        OK = 0;
        if(matrix[ii][j]>0) /* si el elemento es diferente de cero */
        {
            OK = 1;
            /* checa si ya se tiene el vertice */

```

```

    for(k=1;k<=m;k++) if(T[k]==j) noesta = 1;

    /* si el vertice no esta repetido */

    if(noesta==0)
    {
        reg = reg + 1;
        T[reg] = j; /* agrega el numero de columna */
        M[reg] = T[reg-1]; /* agrega el numero de renglon */
    }
}
ii = ii + 1;
if(ii>m) ii = 1;
}
reg = 0;
for(i=1;i<=m;i++)
{
    OK = 0;
    for(j=1;j<=m;j++)
    {
        if(T[i]==M[j] && OK==0)
        {
            reg = reg + 1;
            O[reg] = T[i];
            OK = 1;
        }
    }
}
reg = reg + 1;
O[reg] = T[m];

/* Algoritmo ( P/S/D ) - 5 */

printf("\t\tLa Ruta buscada es:\n\n");
printf("\tN = {");

for(i=1;i<=reg;i++)
{
    if(N[i]==0) continue;
    printf(" V%d ",N[i]);
}
printf("\n\n");
if(T[1]>1)
{
    printf("\t\tOtra Ruta es:\n\n");
    printf("\tM = {");

    for(i=1;i<=reg;i++)
    {
        if(O[i]==0) continue;
        printf(" V%d ",O[i]);
    }

    printf("\n\n");
}
}
else
    printf("\n\n\t\tERROR el archivo de datos ADJACEN.TXT no existe !!\n\n");
}

```

```

/*
*****
Algoritmo ( P/B/D )
Encuentra: Ruta de Vp a Vq
Dado: Matriz de Incidencias B para una gráfica G ( V, A )
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 22 - MAR - 1997
*****
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int m,n,noesta,ligacon;
int i,j,ii,k,reg,pivote,reg1;
char dummy[80];

int matrix[10][10]; /* Matriz de Incidencia */
int T[12]; /* Vertices Checados */
int P[12]; /* Vertices Precedentes */
int N[12]; /* Ruta Buscada */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("inciden.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\\n]\\n",dummy);

        /* Lee el no. de vertices , no. de tramos y el vertice inicial */

        fscanf(a1,"%d %d\\n",&m,&n);

        /* Lee los elementos de la Matriz Incidencia B */

        for(i=1;i<=m;i++)
            for(j=1;j<=n;j++)
            {
                fscanf(a1,"%d",&matrix[i][j]);
            }
        /* Algoritmo que identifica una Ruta apartir de la Matriz de incidencias*/
        /* Paso 1 Asigna volores iniciales */

        reg = 1;

        T[reg] = 1; /* Vertice inicial */
        P[reg] = 0; /* Como ningun vertice precede al V1 se toma el valor de 0 */
        /* Algoritmo ( P/B/D ) - 2 */

        /* Paso 2 Checa cada uno de los renglones y si el elemento es -1 :

                Agrega el numero de Columna a T
                y el tramo a P */

```



```

{
    printf("\tV%d ",i);

    for(j=1;j<=n;j++)
    {
        printf(" %3d ",matrix[i][j]);
    }
    printf("\n");
}
printf("\n");

printf("\t\tVertices Checados\n\n");
printf("\tT = {");

for(i=1;i<=reg;i++)
{
    if(T[i]==0) continue;
    printf(" V%d ",T[i]);
}

printf("}\n\n");

printf("\t\tVertices Precedentes\n\n");
printf("\tP = {");

printf(" 0 ");
for(i=2;i<=reg;i++)
{
    if(P[i]==0) continue;
    printf(" a%d ",P[i]);
}

printf("}\n\n");

printf("\t\tLa Ruta buscada esta compuesta por los arcos :\n\n");
printf("\tN = {");

for(i=0;i<=(reg1-1);i++)
{
    k = reg1 - y;
/* Algoritmo ( P/B/D ) - 4 */

    printf(" a%d ",N[k]);
}

printf("}\n\n");

}
else
    printf("\n\n\t\tERROR el archivo de datos INCIDEN.TXT no existe !!\n\n");
}

```

```

/*
*****
Algoritmo ( PS-1/L/D ) - Dijkstra -
Encuentra: Ruta mas corta de Vp a Vq.
Dado: Matriz de Distancias L de una gráfica dirigida G = ( V, A )
Autor: Ing.TOMAS OLIVER SANCHEZ
Fecha: 22 - MAR - 1997
*****
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int m,n,noesta;
int i,j,k,reg,reg1;
char dummy[80];
int nv,etiqueta,OK,tmp,minimo,col,perma;

int matrix[10][10]; /* Matris de Distancias */
int T[12]; /* Vertices Checados */
int W[12]; /* Total de Vertices */
int P[12]; /* Vertices Pecedentes */
int N[12]; /* Ruta mas corta */
int Pl[12]; /* Vector de Distancias */
int borra[12]; /* Vertices borrados */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("shorpath.txt", "r");
    if(a1 != NULL)
    {
        /* Lee linea de comentario */
        fscanf(a1,"%[^\n]\n",dummy);
        /* Lee el no. de vertices */

        fscanf(a1,"%d\n",&m);

        /* Lee los elementos de la Matriz de Distancias L */

        for(i=1;i<=m;i++)
            for(j=1;j<=m;j++)
            {
                fscanf(a1,"%d",&matrix[i][j]);
            }

        system("cls");
        printf("\n\n\tAlgoritmo de Dijkstra para localizar la Ruta mas corta\n\n");
        printf("\t\t Matriz de Distancias\n\n");
        printf("\t\t ");
    }
    /* Algoritmo ( PS-1/L/D ) - Dijkstra - -2 */

    for(i=1;i<=m;i++) printf(" V%d ",i);
    printf("\n\n");

    for(i=1;i<=m;i++)
    {

```



```

printf("\tV%d ",i);
for(j=1;j<=m;j++)
{
    if(matrix[i][j]==0) printf(" * ");
    else
        printf(" %3d ",matrix[i][j]);
}
printf("\n");
}
printf("\n");
/* Algoritmo de Dijkstra para localizar la ruta mas corta- */

/* Paso 1 Asigna volores iniciales */

reg = 1;
nw = m-1;
etiqueta = 0;
OK = 0; /* continua busqueda */

T[reg] = 1; /* Vertice inicial */
P[reg] = 0; /* Como ningun vertice precede al V1 se toma el valor de 0 */
PL[reg] = 0; /* Vertice inicial */

printf("\tT = { V1 }\n");
printf("\tP = { 0 }\n");
printf("\tPL = { 0 }\n");

/* Guarda todos los vertices */
for(i=1;i<=nw;i++)
{
    W[i] = i+1;
    borra[i] = i+1;
}
printf("\tW = { ");

for(j=1;j<=nw;j++) printf(" V%d ",W[j]);

printf("\n\n");
getch();

/* Paso 2 Usando la Matriz de Distancias L, determina los vertices en W
que son adyacentes a cualquier vertice en T. Asigna a cada uno
de estos vertices una etiqueta temporal igual a la distancia
del vertice desde Vp.

La distancia del vertice Vi en W adyacente a Vj es

/* Algoritmo ( PS-1/L/D ) - Dijkstra - -3 */

temporalmente la nueva etiqueta, dada por:
Etiqueta temporal de Vi = min(etiqueta permanente de Vj+lji). */

for(i=1;i<=m;i++)
{
    if(T[reg]==(m-1)) continue;
    minimo = 10000;
    for(j=1;j<=m;j++) /* encuentra la etiqueta minima por renglon */
    {
        if(OK==1) continue; /* termina busqueda */

```

```

if(matrix[i][j]>0) /* si la distancia es diferente de infinito */
{
    noesta = 0;

    for(k=1;k<=m;k++)
    {
        if(T[k]==j) noesta = 1;
        borra[k] = 0;
    }

    /* si el vertice no esta repetido */

    if(noesta==0)
    {
        /* asigna etiqueta temporal */
        etiqueta = perma + matrix[i][j];

        if(matrix[i][j]<minimo)
        {
            col = j;
            minimo = matrix[i][j];
        }
    }
}
}

```

/* NOTA A partir de la columna donde se encontro la etiqueta minima se continua la busqueda */

```

for(j=col;j<=m;j++) /* continua la busqueda */
{
    if(OK==1) continue; /* termina busqueda */

    if(matrix[i][j]>0) /* si la distancia es diferente de infinito */
    {
        noesta = 0;

        for(k=1;k<=m;k++)
        {
/* Algoritmo ( PS-1/L/D ) - Dijkstra - -4 */

            if(T[k]==j) noesta = 1;
            borra[k] = 0;
        }

        /* si el vertice no esta repetido */

        if(noesta==0)
        {
            /* asigna etiqueta temporal */
            etiqueta = perma + matrix[i][j];

            reg = reg + 1;
            T[reg] = j;          /* agrega el numero de columna */
            P[reg] = i;          /* agrega el numero de renglon */
            PL[reg] = etiqueta; /* etiqueta permanente */
            if(j==m-1) OK = 1; /* fin de la busqueda */

            for(k=1;k<=m;k++) if(W[k]==j) W[k] = 0; /* borra el vertice checado */
        }
    }
}

```

```

        tmp = 1;
        for(k=1;k<=nw;k++) /* copia los vertices pendientes */
        {
            if(W[k]>0)
            {
                borra[tmp] = W[k];
                tmp = tmp + 1;
            }
        }
        for(k=1;k<=m;k++) W[k] = borra[k];
        nw = nw -1 ;
    }
}
printf(" Vertice %d Etiqueta %d\n",col,etiqueta);

perma = etiqueta ;

/* Imprime Resultados */

printf("\tT = {");

for(j=1;j<=reg;j++) printf(" V%d ",T[j]);

printf("\n");

printf("\tP = {");

printf(" 0 ");
for(j=2;j<=reg;j++) printf(" V%d ",P[j]);

printf("\n");

/* Algoritmo ( PS-1/L/D ) - Dijkstra - -5 */
printf("\tPL = {");

for(j=1;j<=reg;j++) printf(" %2d ",PL[j]);

printf("\n");

printf("\tW = {");

for(j=1;j<=nw;j++) printf(" V%d ",W[j]);

printf("\n\n");
getch();
}

reg1 = 0;
for(i=1;i<=reg;i++)
{
    OK = 0;
    for(j=1;j<=reg;j++)
    {
        if(T[i]==P[j] && OK==0)
        {
            reg1 = reg1 + 1;
            N[reg1] = T[i];
            OK = 1;
        }
    }
}
}

```

```

}
reg1 = reg1 + 1;
IJ[reg1] = T[reg];

printf("\tLa Ruta mas corta es :\n\n");
printf("\tN = {");

for(j=1;j<=reg1;j++) printf(" V%d ",N[j]);

printf("\n");

printf("\n");

printf("\ty la distancia de la ruta mas corta es : %d\n\n",etiqueta);
}
else
printf("\n\n\t\tERROR el archivo de datos SHORPATH.TXT no existe !!\n\n");
}

```

```

/*
*****
Algoritmo ( PS-2/L/D )
Encuentra: Ruta mas corta de Vp a Vq.
Dado: Matriz de Distancias de una Gráfica dirigida G = ( V, A )
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 23 - MAR - 1997
*****
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int m,n,noesta;
int i,j,k,reg;
char dummy[80];
int perma,minimo,etiqueta,S,reg1,OK;

int matrix[10][10]; /* Matriz de Distancias */
int W[12]; /* Vertices Checados */
int N[12]; /* Vertices Checados */
int P[12]; /* Vertices Checados */
int T[10][10]; /* Arbol */
int X[12]; /* Etiquetas */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("shortree.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\\n]\\n",dummy);

        /* Lee el no. de vertices */

        fscanf(a1,"%d\\n",&m);

        /* Lee los elementos de la Matriz L de Distancias y crea la Matriz T */
        for(i=1;i<=m;i++)
            for(j=1;j<=m;j++)
            {
                fscanf(a1,"%d",&matrix[i][j]);
                if(i==m || j==m) continue; /* borra el ultimo vertice PASO 1 */
                if(matrix[i][j]>0) T[i][j] = 1; /* crea la Matriz T PASO 2 */
                if(matrix[i][j]==0) T[i][j] = 0;
            }

        system("cls");
        /* Algoritmo ( PS-2/L/D ) - 2 */

        printf("\\n\\n\\tAlgoritmo para identificar la Ruta mas corta\\n\\n");
        printf("\\t\\t Matriz de Distancias\\n\\n");
        printf("\\t ");
        for(i=1;i<=m;i++) printf(" V%d ",i);
        printf("\\n\\n");
    }
}

```

```

for(i=1;i<=m;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        if(matrix[i][j]==0) printf(" * ");
        else
            printf(" %3d ",matrix[i][j]);
    }
    printf("\n");
}
printf("\n");

printf("\t\t Matriz del Arbol de Expansi3n\n\n");
printf("\t\t ");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\n\n");

for(i=1;i<=m;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        printf(" %3d ",T[i][j]);
    }
    printf("\n");
}
printf("\n");

/* Algoritmo para identificar la ruta mas corta */

/* Paso 1 Borra el ultimo vertice */

/* Paso 2 Crea la Matriz del Arbol de expansi3n y asigna etiquetas */

reg = 1;
perma = 0;

W[reg] = 1; /* Vertice inicial */
P[reg] = 1; /* Vertice inicial */
X[reg] = 0; /* Etiqueta inicial */

/* asigna las etiquetas */

for(i=1;i<=m-1;i++)
/* Algoritmo ( PS-2/L/D ) - 3 */
{
    minimo = 10000;

    for(j=1;j<=m-1;j++)
    {
        if(matrix[i][j]>0)
        {
            noesta = 0;
            for(k=1;k<=m;k++) if(P[k]==j) noesta = 1;

            /* si el vertice no esta repetido */

```

```

        if(noesta==0)
        {
            reg = reg + 1;
            etiqueta = matrix[i][j] + perma;
            P[reg] = j;
            W[reg] = i;
            X[reg] = etiqueta;
            if(matrix[i][j]<minimo) minimo = matrix[i][j];
        }
    }
}

perma = etiqueta;
}
printf("\tEl vector de Etiquetas es :\n\n\t");

for(i=1;i<=reg;i++) printf(" X%d = %d ",i,X[i]);
printf("\n\n");

/* PASO 3 Holgura */

for(i=1;i<=m;i++)
    for(j=1;j<=m;j++)
    {
        S = X[j] - X[i] - matrix[i][j];
    }
printf("\tLa Distancia de la Ruta mas corta es : %d\n",X[reg]);
printf("\n");
printf("\tLa Ruta mas corta pasa por los vertices :\n");

reg1 = 0;
for(i=1;i<=reg;i++)
{
    OK = 0;
    for(j=1;j<=reg;j++)
    {
        if(P[i]==W[j] && OK==0)
        {
            reg1 = reg1 + 1;
            N[reg1] = P[i] ;
/* Algoritmo ( PS-2/L/D ) - 4 */

            OK = 1;
        }
    }
}
reg1 = reg1 + 1;
N[reg1] = P[reg];
printf("\n\t");
for(i=1;i<=reg1;i++) printf(" V%d ",N[i]);
printf("\n\n");
}
else
    printf("\n\n\t\t\t\t\tERROR el archivo de datos SHORTREE.TXT no existe !!\n\n");
}

```

```

/*
*****
Algoritmo ( PS-3/L/D )
Encuentra: Encuentra Matriz L+ de distancias más cortas y la Matriz de
          rutas R.
Autor:    Ing. TOMAS OLIVER SANCHEZ
Fecha:    25 - MAR - 1997
*****
*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int i,j,k;
char dummy[80];
int m,pivote,reg,min;

int L[10][10];    /* Matriz de distancias */
int A[10][10];    /* Matriz de adyacencias */
int R[10][10];    /* Matriz de Ruta */
int fil[25];      /* vector de renglones i */
int col[25];      /* vector de columnas k */
int dis[25];      /* vector de distancias lik */
int row[25];      /* vector de columnas rik */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("cascade.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\n]\n",dummy);

        /* Lee el no. de vertices */
        fscanf(a1,"%d\n",&m);
        /* Lee la Matriz L de distancias */

        for(i=1;i<=m;i++)
        {
            for(j=1;j<=m;j++) fscanf(a1,"%d",&L[i][j]);
            fil[i] = 0;
            col[i] = 0;
            dis[i] = 0;
            row[i] = 0;
        }

        system("cls");
        printf("\n\n\t Algoritmo Cascada\n\n");
        printf("\t\t Matriz de Distancias\n\n");
        /* Algoritmo ( PS-3/L/D ) -2 */

        printf("\n\t ");
        for(i=1;i<=m;i++) printf(" V%d ",i);
        printf("\n\n");

        for(i=1;i<=m;i++)

```



```

{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        if(L[i][j]==1000) printf(" * ");
        else
            printf(" %3d ",L[i][j]);
    }
    printf("\n");
}
printf("\n");
/* Lee linea de comentario */
fscanf(a1,"%[^\\n]\\n",dummy);

/* Lee la Matriz A de adyacencias */

for(i=1;i<=m;i++)
    for(j=1;j<=m;j++)
    {
        fscanf(a1,"%d",&A[i][j]);
        R[i][j] = 0;
    }
fclose(a1);

printf("\t\t Matriz Adyacencias\\n\\n");
printf("\t ");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\\n\\n");

for(i=1;i<=m;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        printf(" %3d ",A[i][j]);
    }
    printf("\\n");
}
printf("\\n");

/* Paso 1 Se crea la matriz R */

for(i=1;i<=m;i++)
/* Algoritmo ( PS-3/L/D ) -3 */
for(j=1;j<=m;j++)
{
    if( L[i][j]>0 && L[i][j]<1000) R[i][j] = j;
}

/* Paso 2 Asignan un pivote inicial */

pivote = k;

for(k=1;k<=m;k++) /* Paso 5 repetir los pasos 3 y 4 hasta que pivote = m */
{
    pivote = k;

```

/* Paso 3 Asigna valores a los vectores */

```

reg = 0;
for(i=1;i<=m;i++)
    for(j=1;j<=m;j++)
    {
        if(j!=pivote && i!=pivote)
        {
            if(L[i][j]!=0)
            {
                min = 10000;
                reg = reg + 1;
                fil[reg] = i;
                col[reg] = j;
                if(L[i][j]<min)          dis[reg] = L[i][j];
                if(L[i][j]<min)          min    = L[i][j];
                if( (L[i][pivote]+L[pivote][j])<min )    dis[reg] = L[i][pivote]+L[pivote][j];
                if( L[i][j]<=(L[i][pivote]+L[pivote][j]) ) row[reg] = R[i][j] ;
                else
                    row[reg] = R[i][pivote];
            }
        }
    }
}

```

/* Paso 4 Actualiza la matriz L y R con los elementos encontrados en el paso 3 */

```

reg = 0;
for(i=1;i<=m;i++)
    for(j=1;j<=m;j++)
    {
        if(j!=pivote && i!=pivote)
        {
            if(L[i][j]!=0)
            {
                reg = reg + 1;
                L[i][j] = dis[reg];
                R[i][j] = row[reg];
            }
        }
    }
}

```

/* Algoritmo (PS-3/L/D) -4 */

printf("\n\tLa Distancia de la ruta mas corta esta dada por los elementos de L :\n\n");

printf("\tEjemplo: La distancia de la ruta mas corta\n");

printf("\tdel vertice i a j es L(i,j) y el proximo vertice de la ruta es R(i,j)\n");

printf("\n\n\tMatrix de Distancias L\n\n");

```

printf("\t\t");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\n\n");
for(i=1;i<=m;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        printf(" %3d ",L[i][j]);
    }
    printf("\n");
}

```

```

printf("\n");

printf("\n\n\tMatrix de los vertices de la ruta R\n\n");

printf("\t\t");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\n\n");

for(i=1;i<=m;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        printf(" %3d ",R[i][j]);
    }
    printf("\n");
}
printf("\n");

}
else
    printf("\n\n\t\tERROR el archivo de datos KRUSKAL.TXT no existe !!\n\n");
}

```

```

/*
*****
Algoritmo ( PH/S/D ) - Kaufmann - ( Método de la multiplicacion Latina )
Encuentra: Ruta de Hamilton.
Dado: Matriz de Adyacencias S de una gráfica G ( V, A ) o las
      rutas simples de G.
Autor: Ing. TOMAS OLIVER SANCHEZ.
Fecha: 26 - MAR - 1997
*****

```

```

*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;
FILE *a2;
FILE *a3;

int i,j, <,ii,jj;
char dummy[80];
int ver,m,n,reg,factor,fil,fil1,nofil;
int res,q,con,col;
float t np,suma,ele;

float M[20][20]; /* Matriz M */
float N[20][20]; /* Matriz con vertices de M */
float L[20][20]; /* Multiplicacion Latina */
int P[50]; /* Vertices */

```

```

main()
{
    /* Abre archivo de datos */
    a1 = fopen("latina.txt", "r");
    if(a1 == NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\n]\n",dummy);
        /* Lee el No. de Vertices */
        fscanf(a1,"%d\n",&ver);

        /* Lee la Matriz de Circuitos */

        for(i=1;i<=2*ver;i+=2)
            for(j=1;j<=ver;j++)
            {
                fscanf(a1,"%f",&M[i][j]);
                if(M[i][j]>0) N[i][j] = (float)(j); /* PASO: 2 */
                else
                    N[i][j] = 0;

                L[i][j] = 0;
            }
    }
    /* Algoritmo ( PH/S/D ) -2 */

    fclose(a1);
    system("cls");
    printf("\n\n\tAlgoritmo para obtener la ruta de Hamilton\n");
    printf("\tpor medio de la multiplicacion Latina\n");
    printf("\t\tMatriz Adjunta M(1) :\n");

```

```

printf("\t ");
for(i=1;i<=ver;i++) printf(" %2d ",i);
printf("\n\n");
for(i=1;i<=2*ver;i+=2)
{
    printf("\t%d ",(int)(i/2+1.511111));

    for(j=1;j<=ver;j++)
    {
        printf(" %3.0f ",M[i][j]);
    }
    printf("\n");
}
printf("\n\n");

printf("\t\t Matriz de Vertices N(1) :\n");
printf("\t ");
for(i=1;i<=ver;i++) printf(" %2d ",i);
printf("\n\n");

for(i=1;i<=2*ver;i+=2)
{
    printf("\t%d ",(int)(i/2+1.511111));

    for(j=1;j<=ver;j++)
    {
        printf(" %3.0f ",N[i][j]);
    }
    printf("\n");
}
printf("\n\n");

/* Algoritmo para obtener la ruta de Hamilton */
/* por medio de la multiplicacion Latina */

/* Paso 3 Realiza la multiplicacion Latina */

k = 1;
factor = 1;

do
{
    k = k + 1;
    reg = 0;
    factor = factor * 10;
    for(i=1;i<=2*ver;i++)
    {
/* Algoritmo ( PH/S/D ) -3 */

        suma = 0;
        for(j=1;j<=ver;j++) suma = M[i][j] + suma;
        if(suma==0) continue;
        suma = 0;
        for(j=1;j<=ver;j++)
        {
            for(n=1;n<=2*ver;n++)
            {
                reg = reg + 1;
                col = (int)(n/2+1.511111);
                if(M[i][col]*N[n][j]>0)

```

```

{
  m = M[i][col]/factor;
  if(m!=j)
  {
    tmp = M[i][col]*10 + N[n][j]; /* V1V2 x V5 = V1V2V5 concatena */

    /* V1V2V3V1 = 0 tiene un vertice repetido */
    q = factor;
    con = 0;
    ele = M[i][col];

    /* vertices de la Matrix M */
    for(jj=1;jj<=k;jj++)
    {
      if(ele==0) continue;
      res = (int)(ele/q);
      con = con + 1;
      P[con] = res;
      ele = ele - res*q;
      q = q / 10;
    }

    /* vertices de la Matrix N */
    con = con + 1;
    P[con] = N[n][j];

    /* Si hay vertices repetidos los borra */
    for(jj=1;jj<=con-1;jj++)
    {
      q = jj;
      do
      {
        q = q + 1;
        if(P[jj]==P[q]) tmp = 0;
      }
      while(q<con);
    }

    if(tmp>0 && suma>0) /* V1V2 + V6V7 = V1V2 */
/* Algoritmo ( PH/S/D ) -4 */
    {
      /* V6V7 se agrega un renglon */
      L[i+1][j] = tmp;
    }
    if(suma==0 || tmp==0) suma = suma + tmp;
  }
}
L[i][j] = suma;
suma = 0;
tmp = 0;
col = j;
if(fi!=fil1) ii = ii + 1;
}
}
for(i=1;i<=2*ver;i++)
for(j=1;j<=ver;j++)
{
  M[i][j] = L[i][j];
  L[i][j] = 0;
}

```

```

/*      for(i=1;i<=fil;i++)
        for(j=1;j<=ver;j++)
        {
            suma = (int)(M[i][j]/(factor*10));
            N[i][j] = M[i][j] - suma*factor*10;
        } */

if(k<ver)
{
    a2 = fopen("borra.txt" , "w");
    printf("\tPASO 3:\n");
    printf("\tMatriz M(%d):\n",k);
    printf(" ");
    for(i=1;i<=ver;i++) printf(" %6d",i);
    printf("\n");

    reg = 0;
    for(i=1;i<=2*ver;i++)
    {
        fil = (int)(i/2);
        if(i>2*fil) reg = reg + 1;

        if(i>2*fil) printf("\t%2d ",reg);
        else
            printf("\t  ");

        for(j=1;j<=ver;j++)
        {
            printf(" %6.0f",M[i][j]);
        }
        printf("\n");
    }
/* Algoritmo ( PH/S/D ) -5 */

    }
    printf("\n\n");
    fclose(a2);

    printf("\tMatriz N(1):\n");
    printf("\t  ");
    for(i=1;i<=ver;i++) printf(" %2d ",i);
    printf("\n\n");

    for(i=1;i<=2*ver;i+=2)
    {
        printf("\t%d ",(int)(i/2+1.511111));

        for(j=1;j<=ver;j++)
        {
            printf(" %3.0f ",N[i][j]);
        }
        printf("\n");
    }
    printf("\n\n");
}
}
while(k<ver);
}
else
    printf("\n\n\t\tERROR el archivo de datos KRUSKAL.TXT no existe !!\n\n");
}

```

```

/*
*****
Algoritmo ( PL/L/D )
Encuentra: Ruta mas larga
Dado: Matriz de distancias L de una gráfica dirigida G = ( V, A )
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 23 - MAR - 1997
*****

*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int paso2();
int paso3();

int m,n,noesta;
int i,j,c,reg;
char dummy[80];
int etiqueta,S,negativo,col,fil,max,reg1,OK;

int matrix[10][10]; /* Matriz de Distancias */
int T[12]; /* Vertices checados */
int P[12]; /* Vertices precedente */
int N[12]; /* Ruta */
int PL[12]; /* Vector de Distancias */
int TREE[10][10]; /* Arbol */
int X[12]; /* Etiquetas */

main()
{
/* Abre archivo de datos */
a1 = fopen("longtree.txt", "r");
if(a1 == NULL)
{
/* Lee linea de comentario */
fscanf(a1,"%[^\n]\n",dummy);
/* Lee el no. de vertices */

fscanf(a1,"%d\n",&m);

/* Lee los elementos de la Matriz L de Distancias */

for(i=1;i<=m;i++)
for(j=1;j<=m;j++)
{
fscanf(a1,"%d",&matrix[i][j]);
}

/* Lee linea de comentario */

/* Algoritmo ( PL/L/D ) -2 */

fscanf(a1,"%n");
fscanf(a1,"%[^\n]\n",dummy);
/* Lee los elementos del Arbol T */

for(i=1;i<=m;i++)

```



```

        for(j=1;j<=m;j++)
        {
            fscanf(a1,"%d",&TREE[i][j]);
        }

system("cls");
printf("\n\n\t\t\tAlgoritmo para identificar la Ruta mas larga\n\n");
printf("\t\t\t\t\tMatriz de Distancias\n\n");
printf("\t\t\t\t\t");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\n\n");

for(i=1;i<=m;i++)
{
    printf("\t\t\t\t\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        if(matrix[i][j]==0) printf(" * ");
        else
            printf(" %3d ",matrix[i][j]);
    }
    printf("\n");
}
printf("\n\n");
printf("\t\t\t\t\tRaíz del Árbol T T\n\n");
printf("\t\t\t\t\t");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\n\n");

for(i=1;i<=m;i++)
{
    printf("\t\t\t\t\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        printf(" %3d ",TREE[i][j]);
    }
    printf("\n");
}
printf("\n\n");

/* Algoritmo para identificar la Ruta mas larga */

/* Paso 1 La Matriz del Arbol es obtenida de la Matrix T */

/* Paso 2 Crea la Matriz del Arbol de expansión y asigna etiquetas */
/* Algoritmo ( PL/L/D ) -3 */

paso2();
/* PASO 3 Slack */

paso3();

if(negativo == 1)
{
    do
    {
        printf("\n\n\t\t\tfila %d columnas %d\n\n",fil,col);
        /* PASO 4 Iguala las columnas de la Matriz T a 0 */
        printf("\t\t\t\t\tPASO 4 Columna pivote %d\n\n",col);
    }
}

```

```

    for(i=1;i<=m;i++)
    {
        TREE[i][col] = 0;
        if(i==fil) TREE[fil][col] = 1;
    }

    /* Repite paso 2 y 3 hasta no haya negativos */

    printf("\tPASO 2\n");
    paso2();

    printf("\tPASO 3\n");
    paso3();

    getch(); /* pausa */
}
while(negativo!=0);
}
rmax = -15;
for(i=1;i<=m;i++)
{
    if(X[i]>max)
    {
        max = X[i];
        fil = i;
    }
}
reg1 = 1;
N[1] = 1;
for(i=1;i<=reg;i++)
{
    OK = 0;
    for(j=1;j<=reg;j++)
    {
        if(T[i]==P[j] && OK==0)
        {
            reg1 = reg1 + 1;
        }
    }
}

/* Algoritmo ( PL/LD ) -4 */

    N[reg1] = T[i] ;
    OK = 1;
}
}
reg1 = reg1 + 1;
N[reg1] = T[reg];

printf("\tLa Ruta mas larga pasa por los vertices \n\n");
printf("\tN = {");
for(j=1;j<=reg1;j++) printf(" V%d ",N[j]);

printf("\n");

printf("\n");

printf("\n\t La ruta mas larga tiene un valor de X%d = %d\n\n",fil,max);
}

```

```

else
    printf("\n\n\t\tERROR el archivo de datos LONGTREE.TXT no existe !!\n\n");
}
int paso2()
{
    /* valores iniciales */
    reg = 0;
    reg1 = 0;

    T[reg+1] = 1;
    N[reg1+1] = 1;
    P[reg+1] = 0;
    PL[reg+1] = 0;
    X[reg+1] = 0;

    /* asigna las etiquetas */
    for(i=1;i<=m;i++)
        for(j=1;j<=m;j++)
        {
            if(matrix[i][j]>0 && TREE[i][j]>0)
            {
                /* si el tree es mayor que cero */
                reg = reg + 1;
                etiqueta = matrix[i][j] + PL[i];
                T[reg] = j;          /* agrega el numero de columna */
                P[reg] = i;          /* agrega el numero de renglon */
                PL[j] = etiqueta;    /* etiqueta permanente */
                X[j] = etiqueta;
            }
        }

    /* Algoritmo ( PL/L/D ) -5 */
    printf("\t");
    for(i=1;i<=m;i++) printf(" X%d = %d ",i,X[i]);
    printf("\n\n");
    return;
}
int paso3()
{
    negativo = 0;

    printf("\t");
    for(i=1;i<=m;i++)
        for(j=1;j<=m;j++)
        {
            if(matrix[i][j]>0)
            {
                S = X[j] - X[i] - matrix[i][j];
                if(S!=0) printf(" S%d%d = %d ",i,j,S);
                if(S<0)
                {
                    negativo = 1;
                    fil = i;
                    col = j;
                }
            }
        }
    printf("\n\n");

    return;
}

```

```

/*
*****
Algoritmo ( C/B/U )
Encuentra: Matriz de Circuitos Básicos
Dado: Matriz de Incidencias B
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 25 - MAR - 1997
*****
*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int i,j,k;
char dummy[80];
int m,n,pivote,reg,x,p,OK;

int B[10][10]; /* Matriz de incidencias */
int M[10][10]; /* Matriz de Circuitos Básicos */
int F[25]; /* Elementos */
int C[25]; /* Columnas que no han sido borradas */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("bcircuit.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\n]\n",dummy);

        /* Lee el no. de vertices */

        fscanf(a1,"%d %d\n",&m,&n);
        /* Lee la Matriz de Incidencias */

        for(i=1;i<=m;i++)
        {
            for(j=1;j<=n;j++)
            {
                fscanf(a1,"%d",&B[i][j]);
            }
            F[i] = 0;
        }
        system("cls");
        printf("\n\n\tAlgoritmo para obtener un Circuito Base\n");
        printf("\ta partir de la Matriz de Incidencias\n\n");
        printf("\t\tMatriz de Incidencias\n\n");
        printf("\t\t");
        for(i=1;i<=m;i++) printf(" V%d ",i);
        /* Algoritmo ( C/B/U ) - 2 */

        printf("\n\n");
        for(i=1;i<=m;i++)
        {
            printf("\tV%d ",i);

```

```

    for(j=1;j<=n;j++)
    {
        printf(" %3d ",B[i][j]);
    }
    printf("\n");
}
printf("\n");

/* Algoritmo para obtener una matriz de circuitos basicos */

/* Paso 1 Dar el arbol de expansion T */

reg = 0;
pivote = 1;
for(j=1;j<=n;j++)
{
    if(B[pivote][j]==0)
    {
        reg = reg + 1;
        F[reg] = j; /* Paso 2 Elige los elementos = 0 del renglon pivote */
    }
    C[j] = j;
}

do
{
    /* Paso 3 Se elige el primero y se borran las columnas asociadas a los demas */

    printf("\t\tPivote %d\n",pivote);
    for(i=1;i<=reg;i++)
        for(j=1;j<=m;j++)
        {
            if(F[pivote]==F[i]) continue;

            B[j][F[i]] = 0;
            C[F[i]] = 0;
        }

    /* Paso 4 Borra primera columna donde el elemento == 1 y renglon F[pivote] */
    if(F[pivote]>m) F[pivote] = pivote;
    OK = 0;
    for(j=1;j<=n;j++)
    {
        if(B[F[pivote]][j]>0 && j<n && OK==0) /* borra primer columna */
        {
            printf("\tBorra columna %d\n",j);
            for(i=1;i<=m;i++)
                B[i][j] = 0;
        }
    }
} while (OK==0);

/* Algoritmo ( C/B/U ) - 3 */

{
    B[i][j] = 0;
}
OK = 1;
C[j] = 0;
printf("\tBorra renglon pivote %d\n",pivote);
}
B[F[pivote]][j] = 0; /* borra renglon F[pivote] */
}

/* Paso 5 Asigna el valor de 1 a toda columna que no ha sido borra */

printf("\t");

```

```

for(i=1;i<=n;i++)
{
    if(C[i]==i)
    {
        M[pivote][i] = 1;
        printf(" C%d%d ",pivote,i);
    }
}
printf(" = 1 \n\n");

/* Paso 6 Asigna pivote = pivote +1; si p = x+2 para si no repite el paso 3-6 */

pivote = pivote + 1;
x = n - m;
for(i=1;i<=n;i++) C[i]=i;

}
while(pivote<(x+2));

printf("\n\tLa Matriz del Circuitos Básicos es :\n\n");

printf("\t ");
for(i=1;i<=n;i++) printf(" E%d ",i);
printf("\n\n");

for(i=1;i<=pivote-1;i++)
{
    printf("\tC%d ",i);

    for(j=1;j<=n;j++)
    {
        printf(" %3d ",M[i][j]);
    }
    printf("\n");
}
printf("\n");

}
/* Algoritmo ( C/B/U ) - 4*/

else
    printf("\n\n\t\tERROR el archivo de datos KRUSKAL.TXT no existe !!\n\n");
}

```

```

/*
*****
Algorithmo ( CE/B/U )
Encuentra: Circuito de Euler.
Dado: Matriz de Incidencias B de una Gráfica G ( V, E ) sin rizos.
Autor: Ing. TOMAS OLIVER SANCHEZ.
Fecha: 26 - MAR - 1997
*****
*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int i,j,k;
char dummy[80];
int m,n,suma,OK,fil,col,pivote;

int B[10][10]; /* Matriz de Incidencias */
int M[25]; /* Matriz de Circuitos */
int V[25]; /* Vertices */
int E[25]; /* Elementos */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("circuiti.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */
        fscanf(a1,"%[^\\n]\\n",dummy);
        /* Lee el no. de vertices */

        fscanf(a1,"%d %d\\n",&m,&n);

        /* Lee la Matriz Incidencias */

        for(i=1;i<=m;i++)
        {
            for(j=1;j<=n;j++)
            {
                fscanf(a1,"%d",&B[i][j]);
            }
            V[i] = 0;
            E[i] = 0;
        }

        /* Algoritmo para localizar un circuito de Euler */

        /* Paso 1 Checa si existe el Circuito de Euler */
        OK = 0;

/* Algorithmo ( CE/B/U ) -2 */

        for(i=1;i<=m;i++)
        {
            suma = 0;
            if(OK == 1) continue;
            for(j=1;j<=n;j++)

```

```

    {
        suma = suma + B[i][j];
    }
    if(suma==0) OK = 1;
}

if(OK==1) printf("\n\n\tNo existe el circuit de Euler !!\n\n");
if(OK==1) exit(1);
pivote = 0;

do
{
    /* Paso 2 Incrementa pivote, Inicia en una columna k diferente de cero,
       y encuentra el circuito de la siguiente form: n = k */

    pivote = pivote + 1;
    OK = 0;
    for(i=1;i<=m;i++)
        for(j=1;j<=n;j++)
        {
            if(OK==1) continue;
            if(B[i][j]!=0) k = j; /* columna de partida */
            if(B[i][j]!=0) OK = 1;
        }
    do
    {
        printf(">>> pivote %d\n",k);
        OK = 0;
        for(i=1;i<=n;i++)
        {
            if(OK==1) continue;
            if(B[k][i]==1) col = i; /* encuentra columna */
            if(B[k][i]!=0) OK = 1;
        }
        OK = 0;
        for(i=1;i<=m;i++)
        {
            if(OK==1) continue;
            if(B[i][col]==1) fil = i; /* encuentra renglon */
            if(B[i][col]!=0) OK = 1;
        }

        printf(">>> fil %d col %d\n",fil,col);
        M[pivote] = B[fil][col];
        for(i=1;i<=m;i++)
            for(j=1;j<=n;j++)
            {
                /* Algoritmo ( CE/B/U ) -3 */

                if(col==j) B[i][j] = 0;
            }

            if(col!=k) k = col;
        }
    while(col!=k);

    OK = 0;
    for(i=1;i<=m;i++)
        for(j=1;j<=n;j++)
        {
            if(B[i][j]>0)

```



```

/*
*****
Algoritmo ( CE/S/U )
Encuentra: Circuito de Euler
Dado: Matriz de Adyacencias S de una gráfica no dirigida G ( V, E )
      sin rizados
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 26 - MAR - 1997
*****

```

```

*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int i,j,k;
char dummy[80];
int m,n,suma,OK,p,reg,ver,fil,inicio,tmp;
int S[10][10]; /* Matriz de Adyacencias */
int C[10][10]; /* Matriz de Circuitos */
int N[26]; /* Vertices de Circuitos */
int M[26]; /* Vertices de Circuito */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("circuita.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */
        fscanf(a1,"%[^\n]\n",dummy);

        /* Lee el no. de vertices */

        fscanf(a1,"%d\n",&ver);

        /* Lee la Matriz de Adyacencias */

        for(i=1;i<=ver;i++)
            for(j=1;j<=ver;j++)
            {
                fscanf(a1,"%d",&S[i][j]);
                C[i][j] = 0;
            }

        system("cls");
        printf("\n\n\tAlgoritmo para localizar un Circuito de Euler\n");
        printf("\ta partir de la Matriz de Adyacencias\n\n");
        printf("\t\t\tMatriz Adyacencias\n\n");
        printf("\t\t\t");
        for(i=1;i<=ver;i++) printf(" V%d ",i);
        /* Algoritmo ( CE/S/U ) -2 */

        printf("\n\n");
        for(i=1;i<=ver;i++)
        {
            printf("\tV%d ",i);

```

```

    for(j=1;j<=ver;j++)
    {
        printf(" %3d ",S[i][j]);
    }
    printf("\n");
}
printf("\n\n");

/* Algoritmo para localizar un Circuito de Euler */

/* Paso 1 Checa si existe el Circuito de Euler */
OK = 0;

for(i=1;i<=ver;i++)
{
    suma = 0;
    if(OK == 1) continue;
    for(j=1;j<=ver;j++)
    {
        suma = suma + S[i][j];
    }
    if(suma==0) OK = 1;
}

if(OK==1) printf("\n\n\tNo existe el circuit de Euler !!\n\n");
if(OK==1) exit(1);

reg = 0;
p = 1; /* contador */
k = 1; /* se propone un renglon inicial */
inicio = k;
fil = k;
n = k;
OK = 0;
for(j=1;j<=ver;j++)
{
    if(OK==1) continue;
    if(S[fil][j]>0) inicio = j; /* encuentra renglon inicio */
    if(S[fil][j]!=0) OK = 1;
}
reg = reg + 1;
C[p][reg] = inicio;
OK = 0;
for(j=1;j<=ver;j++)
{
    if(OK==1) continue;
    if(S[fil][j]>0) n = j; /* encuentra renglon */
}
/* Algoritmo ( CE/S/U ) -3 */

if(S[fil][j]!=0) OK = 1;
}
fil = n;

do
{
    /* Paso 2 Incrementa p, Inicia en una columna k diferente de cero,
    y encuentra el circuito de la siguiente forma: col = k */
    do
    {
        OK = 0;
        for(j=1;j<=ver;j++)

```

```

{
    if(OK==1) continue;
    if(S[fil][j]>0) m = j; /* encuentra renglon */
    if(S[fil][j]!=0) OK = 1;
}
reg = reg + 1;
C[p][reg] = m;
S[n][m] = 0;
S[m][n] = 0;

n = m;
fil = m;
}
while(C[p][reg]!=inicio);

printf("\t ");
for(i=1;i<=ver;i++) printf(" V%d ",i);
printf("\n\n");

for(i=1;i<=ver;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=ver;j++)
    {
        printf(" %3d ",S[i][j]);
    }
    printf("\n");
}
printf("\n\n");

printf("\tCircuito de Euler C%d\n\n",p);
printf("\tC%d = {",p);

for(i=1;i<=ver;i++)
{
    if(C[p][i]==0) continue;
    printf(" V%d ",C[p][i]);
}

```

/* Algoritmo (CE/S/U) -4 */

```

printf("\n\n");

OK = 0;
for(i=1;i<=ver;i++)
    for(j=1;j<=ver;j++)
    {
        if(S[i][j]>0)
        {
            OK = 1;
        }
    }
p = p + 1;
reg = 0;

/* Continua Paso 2 */

OK = 0;
for(i=1;i<=ver;i++)
    for(j=1;j<=ver;j++)

```

```

    {
        if(OK==1) continue;
        tmp = (int)(p/2);
        if(S[i][j]!=0) OK = 1;
        if(S[i][j]>0) n = j; /* encuentra renglon */
        if(S[i][j]>0 && 2*tmp!=p ) n = i; /* encuentra renglon */
    }
    OK = 0;
    reg = reg + 1;
    i=(int)(p/2);
    if(p!=2*i) C[p][reg] = fil;
    else
        C[p][reg] = n;

    inicio = C[p][reg];
    k = n;
    if(p!=2*i) fil = fil;
    else
        fil = k;

    suma = 0;
    for(i=1; i<=ver; i++)
        for(j=1; j<=ver; j++)
        {
            suma = suma + S[i][j];
        }
}
while(suma>0); /* Paso 3 Hasta que todos los elemntos de S sean 0 */
p = p -1;

/* Paso 4 */

printf("\n\t\tCircuitos de Euler\n\n");
/* Algoritmo ( CE/S/U ) -5 */

for(i=1; i<=p; i++)
{
    printf("\tC%d = {", i);
    for(j=1; j<=ver; j++)
    {
        if(C[i][j]==0) continue;
        printf(" V%d ", C[i][j]);
    }
    printf(")\n");
}
printf("\n\n");

for(i=1; i<=p; i++)
{
    tmp = 0;
    for(j=1; j<=ver; j++)
    {
        if(C[i][j]==0) continue;
        if(j>tmp)
        {
            tmp = j;
            fil = i;
        }
    }
}
printf("\n\tAdicionando los circuitos que tienen elementos comunes:\n\n");

```

```

printf("\tCircuito Base C%d\n",fil);

for(j=1;j<=tmp;j++) N[j]= C[fil][j];
V[1] = C[fil][1];

reg = 1;
suma = 0;
i = 0;

do
{
    i = i + 1;
    if(i>p) i=1;
    if(i==fil) continue;
    OK = 1;

    for(k=2;k<=tmp;k++)
    {
        if(C[i][1]==N[k]) OK = 0;
    }

    if(OK==1) continue;

    for(k=2;k<=tmp;k++)
    {
        if(C[i][1]==N[k] && OK==0)
/* Algoritmo ( CE/S/U ) -6 */

        {
            OK = 1;
            printf("\tAdiciona Circuito C%d\n",i);
            for(j=1;j<=ver;j++)
            {
                if(C[i][j]==0) continue;
                reg = reg + 1;
                M[reg] = C[i][j];
                C[i][j] = 0;
            }
            suma = suma + 1;
        }
        else
        {
            reg = reg + 1;
            M[reg] = N[k];
        }
    }
    :
    tmp = reg;
    reg = 1;
    OK = 0;
    for(k=1;k<=tmp;k++) N[k] = M[k];
    printf("\n");

    printf("\tM = {");
    for(j=1;j<=tmp;j++)
    {
        printf(" V%d ",M[j]);
    }
    printf("\n");
    printf("\n");
}
while(suma<(p-1));

```

```

printf("\n");

printf("\n\tEl Circuito de Euler lo compone el siguiente vector:\n\n");
printf("\tM = {");
for(j=1;j<=tmp;j++)
{
    printf(" V%d ",M[j]);
}
printf(")\n\n\n");

}
else
    printf("\n\n\t\tERROR el archivo de datos KRUSKAL.TXT no existe !!\n\n");
}

```

```

/*
*****
Algoritmo ( CE/S/D )
Encuentra: Circuito de Euler.
Dado: Matriz S de Adyacencias de una Gráfica G ( V, A ) sin rizos.
Autor: Ing. TOMAS OLIVER SANCHEZ.
Fecha: 26 - MAR - 1997
*****
*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int i,j,k;
char dummy[80];
int m,n,suma,OK,p,reg,ver,inicio,tmp,fil;

int S[10][10]; /* Matriz de Adyacencias */
int C[10][10]; /* Ciclos de Euler */
int CC[26]; /* Vector de suma de columnas de S */
int RCW[26]; /* Vector de suma de renglones de S */
int N[26]; /* Vertices del Ciclo de Euler */
int M[26]; /* Vertices del Ciclo de Euler */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("ecycle.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\n]\n",dummy);

        /* Lee el no. de vertices */

        fscanf(a1,"%d\n",&ver);
        /* Lee la Matriz de Adyacencias */

        for(i=1;i<=ver;i++)
            for(j=1;j<=ver;j++)
            {
                fscanf(a1,"%d",&S[i][j]);
                C[i][j] = 0;
            }
        system("cls");
        printf("\n\n\tAlgoritmo para identificar un Ciclo de Euler\n");
        printf("\ta partir de la Matriz de Adyacencias\n\n");
        printf("\tt\t Matriz de Adyacencias\n");
        printf("\t\t");
        for(i=1;i<=ver;i++) printf(" V%d ",i);
        /* Algoritmo ( CE/S/D ) -2 */

        printf("\n\n");
        for(i=1;i<=ver;i++)
        {
            printf("\tV%d ",i);

```



```

    for(j=1;j<=ver;j++)
    {
        printf(" %3d ",S[i][j]);
    }
    printf("\n");
}
printf("\n\n");

/* Algoritmo para identificar un Ciclo de Euler */

/* Paso 1 Checa si existe el Ciclo de Euler */

for(i=1;i<=ver;i++)
    for(j=1;j<=ver;j++)
    {
        ROW[i] = ROW[i] + S[i][j];
        COL[j] = COL[j] + S[j][i];
    }

OK = 0;
for(i=1;i<=ver;i++)
{
    if(ROW[i]!=COL[i]) OK = 1 ;
}

if(OK==1) printf("\n\n\tNo existe Ciclo de Euler !!\n\n");
if(OK==1) exit(1);

/* Paso 2 Incrementa p, Buscar para cada renglon en todas
las columnas de S, hasta encontrar Sij>0 */

reg = 0;
p = 1; /* contador */
k = 1; /* se propone un renglon inicial */
inicio = k;
fil = k;

OK = 0;
for(i=1;i<=ver;i++)
    for(j=1;j<=ver;j++)
    {
        if(OK==1) continue;
        if(S[i][j]>0) inicio = j; /* encuentra renglon inicio */
        if(S[i][j]!=0) OK = 1;
    }
    fil = inicio;

/* Algoritmo ( CE/S/D ) -3 */

OK = 0;
for(j=1;j<=ver;j++)
{
    if(OK==1) continue;
    if(S[fil][j]>0) k = j; /* encuentra columna k */
    if(S[fil][j]!=0) OK = 1;
}
inicio = k;
fil = k;
n = k;
reg = reg + 1;
C[p][reg] = k;

```

```

OK = 0;
for(j=1;j<=ver;j++)
{
    if(OK==1) continue;
    if(S[fil][j]>0) m = j; /* encuentra columna m */
    if(S[fil][j]!=0) OK = 1;
}
reg = reg + 1;
C[p][reg] = m;
S[n][m] = 0;
S[m][n] = 0;
n = m;

do
{
    /* Continua Paso 2 */
    do
    {
        OK = 0;
        for(j=1;j<=ver;j++)
        {
            if(OK==1) continue;
            if(S[n][j]>0) m = j; /* encuentra renglon */
            if(S[n][j]!=0) OK = 1;
        }
        reg = reg + 1;
        C[p][reg] = m;

        S[n][m] = 0;
        S[m][n] = 0;

        n = m;
        fil = m;
    }
    while(C[p][reg]!=inicio);

    printf("\t ");
    for(i=1;i<=ver;i++) printf(" V%d ",i);
    printf("\n\n");
}

```

/* Algoritmo (CE/S/D) -4 */

```

for(i=1;i<=ver;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=ver;j++)
    {
        printf(" %3d ",S[i][j]);
    }
    printf("\n");
}
printf("\n\n");

printf("\tCircuito de Euler C%d\n\n",p);
printf("\tC%d = {",p);

for(i=1;i<=ver;i++)
{
    if(C[p][i]==0) continue;
    printf(" V%d ",C[p][i]);
}

```

```

    }
    printf("\n\n");

    p = p + 1;
    reg = 0;
    reg = reg + 1;
    C[p][reg] = m;
    inicio = m;
    n = m;

    suma = 0;
    for(i=1;i<=ver;i++)
        for(j=1;j<=ver;j++)
        {
            suma = suma + S[i][j];
        }
    }
    while(suma>0); /* Paso 3 Hasta que todos los elemntos de S sean 0 */
    p = p - 1;

    /* Paso 4 */

    printf("\n\t\tCircuitos de Euler\n\n");
    for(i=1;i<=p;i++)
    {
        printf("\tC%d = {", i);
        for(j=1;j<=ver;j++)
        {
            if(C[i][j]==0) continue;
            printf(" V%d ", C[i][j]);
        }
        printf("\n");
    }
}
/* Algoritmo ( CE/S/D ) -5 */

printf("\n\n");

for(i=1;i<=p;i++)
{
    tmp = 0;
    for(j=1;j<=ver;j++)
    {
        if(C[i][j]==0) continue;
        if(j>tmp)
        {
            tmp = j;
            fil = i;
        }
    }
}

printf("\n\tAdicionando los circuitos que tienen elementos comunes:\n\n");
printf("\tCircuito Base C%d\n", fil);

for(j=1;j<=tmp;j++) N[j] = C[fil][j];
M[1] = C[fil][1];

reg = 1;
suma = 0;
i = 0;

do

```



```

/*
*****
Algoritmo ( T-1/B/U )
Encuentra: Árbol de expansión
Dado: Matriz de Incidencias B de una gráfica no dirigida G = ( V, E )
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 23 - MAR - 1997
*****
*/

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int m,n,noesta;
int i,j,k,reg;
char dummy[80];
int pivote;

int matrix[10][10]; /* Matriz de Incidencias */
int T[12]; /* Árbol */
int E[12]; /* Etiquetas */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("spantree.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\\n]\\n",dummy);

        /* Lee el no. de vertices y el no. de tramos */

        fscanf(a1,"%d %d\\n",&m,&n);
        /* Lee la Matriz de Incidencias */

        for(i=1;i<=m;i++)
            for(j=1;j<=n;j++)
            {
                fscanf(a1,"%d",&matrix[i][j]);
            }
        system("cls");
        printf("\\n\\n\\tAlgoritmo para identificar el Arbol de expansión\\n");
        printf("\\t\\ta partir de la Matriz de Incidencias\\n\\n");
        printf("\\t\\t Matriz de Incidencias\\n\\n");
        printf("\\t ");
        for(i=1;i<=n;i++) printf(" V%d ",i);
        printf("\\n\\n");

        /* Algoritmo ( T-1/B/U ) -2 */

        for(i=1;i<=m;i++)
        {
            printf("\\tV%d ",i);

            for(j=1;j<=n;j++)

```

```

    {
        printf(" %3d ",matrix[i][j]);
    }
    printf("\n");
}
printf("\n");
/* Algoritmo para identificar un árbol de expansión */
reg = 0;

/* Paso 1 Asigna a un vector T el tramo inicial y etiquetas iniciales */

for(i=1;i<=n;i++)
{
    T[i] = 0;
    E[i] = 0;
}

for(i=1;i<=m;i++)
{
    if(reg==(m-1)) continue;

    /* Paso 2 Incluye el elemento en T y lo borra de E */

    pivote = 0;
    for(j=1;j<=n;j++)
    {
        if(matrix[i][j]>0 && pivote==0)
        {
            printf("\tincluye el elemento %d en T y lo borra de E\n",i);
            reg = reg + 1;
            T[reg] = i;
            E[i] = -1; /* borra el elemento */
            pivote = 1;
        }
    }
    /* PASO 3 En cada vertice checado agrega 1 en la etiqueta,
    en todo elemento diferente de cero */
    for(j=1;j<=n;j++)
    {
        if(matrix[i][j]>0 && E[j]==0)
        {
            E[j] = 1;
        }
    }

    /* PASO 4 Borra todos los elementos etiquetados, si no tiene etiqueta
    los iguala a 1 y se va al Paso 5,
    /* Algoritmo ( T-1/B/U ) -3 */

    de otra manera selecciona otro elemento con etiqueta igual a 1 */

    for(j=1;j<=n;j++) E[j] = 0;
}

/* PASO 5 Cuando se tenga (n-1) Elementos en T,
    entonces tenemos en T la ruta requerida */

printf("\n\t\tEl Arbol de Expansion es :\n\n");
printf("\tT = {");
for(j=1;j<=reg;j++)
{

```

}

```

/*
****
Algoritmo ( T-2/S/U )
Encuentra: Arbol de Expansi3n
Cado: Matriz de Adyacencias S de una gr3fica G ( V, E )
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 23 - MAR - 1997
****

*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int m,n,noesta;
int i,j,k,reg;
char dummy[80];
int suma;
int matrix[10][10]; /* Matriz de Distancias */
int T[10][10]; /* 3rbol */
int P[12]; /* Conecci3n */

main()
{
    /* Abre archivo de datos */
    a1 :: fopen("findtree.txt" , "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\n]\n",dummy);

        /* Lee el no. de vertices y a partir de que vertice empieza a buscar */

        fscanf(a1,"%d %d\n",&m,&n);

        /* Lee la Matriz Adyacencias */

        for(i=1;i<=m;i++)
            for(j=1;j<=m;j++)
            {
                fscanf(a1,"%d",&matrix[i][j]);
                T[i][j] = 0; /* PASO 1 */
                P[j] = 0;
            }

        system("cls");
        printf("\n\n\tAlgoritmo para encontrar el Arbol de expansi3n\n");
        printf("\ta partir de la Matriz de Adyacencias\n\n");
        printf("\t\tMatriz de Adyacencias\n\n");
        printf("\t ");
        for(i=1;i<=m;i++) printf(" V%d ",i);
        /* Algoritmo ( T-2/S/U ) -2 */

        printf("\n\n");
        for(i=1;i<=m;i++)
        {
            printf("\tV%d ",i);

```



```

        for(j=1;j<=m;j++)
        {
            printf(" %3d ",matrix[i][j]);
        }
        printf("\n");
    }
    printf("\n");

    reg = 0;
    suma = 0;

    /* Algoritmo para encontrar un Arbol de expansión */

    /* Paso 1 Asigna a T vertices checados con valor inicial */

    i = n;
    do
    {
        for(j=1;j<=m;j++)
        {
            if(matrix[i][j]>0 && P[j]!=j) /* Paso 2 Checa el renglon i y borra la columna i */
            {
                /* Paso 3 Checa la columna */
                printf("\t\tCheca la Columna %d y borra el renglon %d\n",i,i);
                suma = suma + 1;
                T[i][j] = 1;
                P[i] = i;
                for(k=1;k<=m;k++) matrix[k][j] = 0; /* borra la columna */
            }
        }
        i = i + 1;
        if(i>m) i = 1;
    }
    while(suma<(m-1));

    printf("\n\t\tEl Arbol de expansión esta dada por la siguiente Matriz:\n\n");
    printf("\t\t");
    for(i=1;i<=m;i++) printf(" V%d ",i);
    printf("\n\n");

    for(i=1;i<=m;i++)
    {
        printf("\tV%d ",i);

        for(j=1;j<=m;j++)
        {
            printf(" %3d ",T[i][j]);
        }
    }
    /* Algoritmo ( T-2/S/U ) -3 */

    printf("\n");
    }
    printf("\n");

}
else
    printf("\n\n\t\tERROR el archivo de datos SPANTREE.TXT no existe !!\n\n");
}

```

```

/*
*****
Algoritmo ( T-3/S/D )
Encuentra: Arbol de Expansi3n T
Dada: Matriz de Adyacencias S de una gr3fica G ( V, E )
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 23 - MAR - 1997
*****
*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;

int m, noesta;
int i, j, k, reg;
char dummy[80];
int suma;
int matrix[10][10]; /* Matriz de Adyacencias */
int T[10][10]; /* 3rbol */
int C[12]; /* V3rtices checados */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("identree.txt", "r");
    if(a1 != NULL)
    {
        /* Lee linea de comentario */
        fscanf(a1, "%[^\n]\n", dummy);

        /* Lee el no. de vertices */

        fscanf(a1, "%d\n", &m);

        /* Lee la Matriz de Adyacencias */

        for(i=1; i<=m; i++)
            for(j=1; j<=m; j++)
            {
                fscanf(a1, "%d", &matrix[i][j]);
                T[i][j] = 0;
            }

        system("cls");
        printf("\n\n\tAlgoritmo para identificar un Arbol de expansi3n\n");
        printf("\ta partir de la Matriz de Adyacencias\n\n");
        printf("\t\tMatriz de Adyacencias\n\n");
        printf("\t\t");
        for(i=1; i<=m; i++) printf(" V%d ", i);
        printf("\n\n");

        /* Algoritmo ( T-3/S/D ) - 2 */

        for(i=1; i<=m; i++)
        {
            printf("\tV%d ", i);

            for(j=1; j<=m; j++)

```

```

        {
            printf(" %3d ",matrix[i][j]);
        }
        printf("\n");
    }
    printf("\n");

    reg = 0;
    suma = 0;

    /* Algoritmo para identificar un Arbol de Expansi3n */

    /* Paso 1 Asigna a T v3rtices checados con valor inicial */

    for(i=1;i<=m;i++)
    {
        C[i] = 0;
    }

    for(i=1;i<=m;i++)
    {
        if( suma==(2*(m-1)) ) continue;

        for(j=1;j<=m;j++)
        {
            if(matrix[i][j]>0 && j>reg) /* Paso 2 Checa el renglon i y borra la columna i */
            {
                printf("\tCheca el renglon %d y borra la columna %d\n",i,i);
                reg = reg + 1;
                suma = suma + 2;
                T[i][j] = 1;
                T[j][i] = 1;
                C[i] = 1; /* PASO 3 chequea v3rtice */
                for(k=1;k<=m;k++) matrix[k][j] = 0; /* borra la columna */
            }
        }
    }

    printf("\n\t\tEl Arbol de expansi3n esta dada por la siguiente Matriz:\n\n");
    printf("\t\t");
    for(i=1;i<=m;i++) printf(" V%d ",i);
    printf("\n\n");

    for(i=1;i<=m;i++)
    {
        printf("\tV%d ",i);
    }

    /* Algoritmo ( T-3/S/D ) - 3 */

    for(j=1;j<=m;j++)
    {
        {
            printf(" %3d ",T[i][j]);
        }
        printf("\n");
    }
    printf("\n");

    }
    else
        printf("\n\n\t\tERROR el archivo de datos SPANTREE.TXT no existe !!\n\n");
    }

```

```

/*
*****
Algoritmo ( TSS-1/L/U ) - Kruskal -
Encuentra: Arbol de expansión más corto.
Dado:      Matriz de distancias L de una gráfica no dirigida G ( V, E )
Autor:     Ing. TOMAS OLIVER SANCHEZ
Fecha:     23 - MAR - 1997
*****
*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;
FILE *a2;

int borra();
int m,noesta;
int i,j,k,reg;
char dummy[80];
int falta,suma;

int matrix[10][10]; /* Matriz de Distancias */
int T[10][10]; /* Árbol */
int E[12]; /* Etiquetas */
int D[20]; /* Distancia */
int fil[25]; /* Renglones de las etiquetas */
int col[25]; /* Columnas de las etiquetas */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("kruskal.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\\n]\\n",dummy);

        /* Lee el no. de vertices y a partir de que vertice empieza a buscar */

        fscanf(a1,"%d\\n",&m);

        reg = 0;
        a2 = fopen("tmp.txt", "w");

        for(i=1;i<=m;i++)
            for(j=1;j<=m;j++)
            {
                fscanf(a1,"%d",&matrix[i][j]);
                T[i][j] = 0; /* PASO 1 */
                E[j] = 0;
                if(j>=i && matrix[i][j]>0)
                    /* Algoritmo ( TSS-1/L/U ) - 2 */
            }

        {
            reg = reg + 1;
            if(matrix[i][j]<10) fprintf(a2,"0%d %d %d\\n",matrix[i][j],i,j);
            else
                fprintf(a2,"%d %d %d\\n",matrix[i][j],i,j);
        }
    }
}

```

```

    }
}
fclose(a2);

system("cls");
printf("\n\n\tAlgoritmo de Kruskal para encontrar el Arbol de expansión\n");
printf("\tmas corto a partir de la Matriz de Distancias\n\n");
printf("\t\t Matriz de Distancias\n\n");
printf("\t\t ");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\n\n");
for(i=1;i<=m;i++)
{
    printf("\tV%d ",i);
    for(j=1;j<=m;j++)
    {
        printf(" %3d ",matrix[i][j]);
    }
    printf("\n");
}
printf("\n");

```

```

system("sort < tmp.txt > tmp1.txt");
a2 = fopen("tmp1.txt", "r");

```

```

for(i=1;i<=reg;i++)
{
    fscanf(a2,"%d %d %d\n",&D[i],&fil[i],&col[i]);
}
fclose(a2);
system("del tmp*.txt");

```

/ Algoritmo para encontrar un árbol de expansión */*

/ Paso 1 Se Asignan valores iniciales a T,E y se ordena el vector D */*

/ Paso 2 Asignan un vertice inicial del vector D */*

```

for(i=1;i<=m;i++)
{
    if(fil[1]==i) E[i] = 1;
    if(col[1]==i) E[i] = 1;
    T[fil[1]][col[1]] = 1;
}

```

/ Algoritmo (TSS-1/L/U) - 3 */*

```

falta = 2;
do
{
    /* Paso 3 Borra la distancia del vertice correspondiente */
    borrar();

    /* Paso 4 Asigna el siguiente vertice. Si las etiquetas de los
    vertices E son cero, toma el valor de 2 */
    for(i=1;i<=m;i++)
    {
        if(E[fil[1]]==0)
        {

```

```

        E[fil[1]] = 2;
        falta = falta + 1;
    }
    if(E[col[1]]==0)
    {
        E[col[1]] = 2;
        falta = falta + 1;
    }
    T[fil[1]][col[1]] = 1;
}
printf("\tAsigna el siguiente vertice  T%d%d = 1\n\n",fil[1],col[1]);
}
while(falta!=m);

borra();

/* Paso 5 Asigna el siguiente vertice. Si las etiquetas de los
   vertices E son cero, toma el valor de 2 */

for(i=1;i<=m;i++)
{
    if(E[i]==2) E[i] = 1;
    T[fil[1]][col[1]] = 1;
}

suma = 0;
for(i=1;i<=m;i++)
    for(j=1;j<=m;j++)
    {
        if(T[i][j]==1) suma = suma + matrix[i][j];
    }

printf("\n\tEl Arbol de expansión mas corto tiene un valor de %d\n",suma);
printf("\ty esta dado por la siguiente Matriz:\n\n",suma);

printf("\t  ");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\n\n");

for(i=1;i<=m;i++)
/* Algoritmo ( TSS-1/L/U ) - 4 */
{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        printf(" %3d ",T[i][j]);
    }
    printf("\n");
}
printf("\n");

}
else
    printf("\n\n\t\tERROR el archivo de datos KRUSKAL.TXT no existe !!\n\n");
}
int borra()
{
    printf("\tBorra la distancia      I%d%d\n",fil[1],col[1]);
    for(i=2;i<=reg;i++)

```

```

/*
*****
Algoritmo ( TSS-2/L/U ) - Solin -
Encuentra: Árbol de expansión más corto
Dado: Matriz L de una gráfica no dirigida G = ( V, E )
Autor: Ing. TOMAS OLIVER SANCHEZ
Fecha: 23 - MAR - 1997
*****
*/
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

FILE *a1;
int borra();
int checa();
int m;
int i,j,k;
char dummy[80];
int fil,col2,col,min,suma,restan,OK,ya;

int matrix[10][10]; /* Matriz de Distancias */
int T[10][10]; /* Árbol */
int E[12]; /* Etiquetas */

main()
{
    /* Abre archivo de datos */
    a1 = fopen("solin.txt", "r");
    if(a1!=NULL)
    {
        /* Lee linea de comentario */

        fscanf(a1,"%[^\n]\n",dummy);

        /* Lee el no. de vertices */

        fscanf(a1,"%d\n",&m);

        /* Lee la Matriz de Distancia */

        for(i=1;i<=m;i++)
            for(j=1;j<=m;j++)
            {
                fscanf(a1,"%d",&matrix[i][j]);
                T[i][j] = 0; /* PASO 1 */
                E[j] = 0;
            }

        system("cls");
        printf("\n\n Algoritmo de Solin para encontrar el Arbol de expansión mas corto\n\n");
        printf("\t\t Matriz de Distancias\n\n");
        printf("\t ");
    }
    /* Algoritmo ( TSS-2/L/U ) - 2 */

    for(i=1;i<=m;i++) printf(" V%d ",i);
    printf("\n\n");

    for(i=1;i<=m;i++)
    {

```

```
{  
    D[i-1] = D[i];  
    fil[i-1] = fil[i];  
    col[i-1] = col[i];  
}  
reg = reg - 1;  
  
return;  
}
```



```

printf("\tV%d ",i);

for(j=1;j<=m;j++)
{
    printf(" %3d ",matrix[i][j]);
}
printf("\n");
}
printf("\n");

/* Algoritmo de Solin para el Arbol de expansión más corto */

/* Paso 1 Se Asignan valores iniciales E y borra la columna 1 */
E[1] = 1;
col = 1;
restan = m-1;
OK = 0;

borra();

i = 1;
do
{
    /* Paso 2 Busca la distancia minima del renglon y columna */
    ya = 0;
    min = 10000;
    for(j=1;j<=m;j++) /* dist. min del renglon */
    {
        if(matrix[i][j]<min && matrix[i][j]>0)
        {
            col = j;
            min = matrix[i][j];
        }
    }
    for(k=1;k<=m;k++) /* dist. min de la columna */
    {
        if(matrix[k][col]<min && matrix[k][col]>0 && E[k]==1)
        {
            i = k;
            min = matrix[k][col];
        }
    }
    if(E[col]==0 && E[i]==1) /* registra la posicion del pivote */
    {
        fil = i;
        col2 = col;
        ya = 1;
    }
} while (TSS-2/L/U) - 3 > 0

chea();

if(E[col]==0 && E[i]==1 && ya==0)
{ /* registra la posicion del pivote, si no se ha registrado aun */
    fil = i;
    col2 = col;
}
printf("\tPivote %d%d\n\n",fil,col2);
if(OK==0) /* copia las distancias minimas a la matriz T */
{
    E[col2] = 1;
}

```

```

        T[fil][col2] = min;
        T[col2][fil] = min;
        borra();
        restan = restan - 1;
    }

    printf("\t ");
    for(k=1;k<=m;k++) printf(" V%d ",k);
    printf("\n\n");
    for(k=1;k<=m;k++) /* visualiza el algoritmo solin, paso a paso */
    {
        printf("\tV%d ",k);
        for(j=1;j<=m;j++)
        {
            printf(" %3d",T[k][j]);
        }
        printf("\n");
    }
    printf("\n");
    getch(); /* PAUSA, hasta que se oprima una tecla */
}
while(restan>0);

/* Paso 3 El árbol de expansión más corto esta dado por la Matriz T */

/* suma las distancias de la diagonal superior que hay en la matriz T */

suma = 0;
for(i=1;i<=m;i++)
    for(j=i;j<=m;j++)
    {
        if(T[i][j]>0) suma = suma + T[i][j];
    }

printf("\n\tEl Arbol de expansión mas corto tiene un valor de %d\n",suma);
printf("\ty esta dada por la siguiente Matriz:\n\n");

printf("\t ");
for(i=1;i<=m;i++) printf(" V%d ",i);
printf("\n\n");
/* Algoritmo ( TSS-2/L/U ) - 4 */

for(i=1;i<=m;i++)
{
    printf("\tV%d ",i);

    for(j=1;j<=m;j++)
    {
        printf(" %3d ",T[i][j]);
    }
    printf("\n");
}
printf("\n");

}
else
    printf("\n\n\t\tERROR el archivo de datos KRUSKAL.TXT no existe !!\n\n");
}
/*****
/* Borra la columna pivote de la matriz de distancias */
*****/

```

```

intorra()
{
    for(k=1;k<=m;k++)
    {
        matrix[k][col] = 0;
    }

    return;
}
/*
*****
Verifica que el punto no haya sido revisado y que el
valor de la distancia del pivote sea la minima
tanto en el renglón como en la columna
*****
*/

int checa()
{
    OK = 0;

    for(k=1;k<=m;k++)
    {
        if(E[col]==1) /* si la columna ya se checo se elige otra */
        {
            for(j=1;j<=m;j++) if(E[j]==0) col = j;
        }
        if(matrix[k][col]<min && matrix[k][col]>0 && E[k]==1)
        {
            /* checa que se la distancia minima en el renglon */
            OK=1;
            min = matrix[k][col];
            i = k;
            printf(">uno> i %d col %d    OK %d  min %d E %d\n",i,col,OK,min,E[i]);
/* Algoritmo ( TSS-2/L/U ) - 5 */

        }
        if(matrix[i][k]<min && matrix[i][k]>0 && E[k]==1)
        {
            /* checa que se tambien la distancia minima en la columna */
            OK=1;

            min = matrix[i][k];
            col = k;
            printf(">dos> i %d col %d    OK %d  min %d E %d\n",i,col,OK,min,E[i]);
        }
    }
}

if(OK==1) /* comprueba los valores del nuevo pivote, solo si se cambio */
{
    for(k=1;k<=m;k++)
    {
        if(E[k]==0 && OK==1) col = k; /* que la columna no haya sido checada */

        OK = 0;
    }
}

```

```

    }
    min = 10000;
    for(k=1;k<=m;k++)
    {
        if(matrix[k][col]<min && matrix[k][col]>0 && E[k]==1)
        {
            /* checa que se la distancia minima en el renglon */
            i = k;
            min = matrix[k][col];
            printf(">tres> i %d col %d    OK %d  min %d E %d\n",i,col,OK,min,E[i]);
        }
    }
}
else
/* Algoritmo ( TSS-2/L/U ) - 6 */
{
    i = col;    /* asigna el nuevo renglon donde se buscara el pivota */
    if(i==m) i = 1; /* si ya es el ultimo renglon que empiece de nuevo */
}
return;
}

```

APENDICE B

B. EJEMPLO.

Con el fin de resaltar la metodología empleada para el análisis del MEBC, con la teoría de gráficas, aquí se presenta un ejemplo simplificado de la gráfica establecida en el capítulo cinco, en donde consideramos cuatro de los seis vértices originales, con respecto a los ejes estructurales del MEBC.

EJES ESTRUCTURALES.

- 1) v_1 : Trinomio Educación-Sector Productivo-Sociedad.
- 2) v_2 : Currículum.
- 3) v_3 : Docencia.
- 4) v_4 : Cultura Organizacional Académica.

1) Contexto.

Del MEBC se consideran únicamente los **ejes estructurales**, los cuales nos proporcionan los elementos básicos para lograr una educación basada en competencias, entorno a estos elementos básicos se realiza el análisis de la gráfica resultante (Figura B.1).

2) Objetivos.

Para este caso simplificado podemos considerar como objetivo primordial el análisis de: Areas de síntesis, Arbol de expansión, Circuitos básicos (o fundamentales) y relaciones y Circuitos Inferidos.

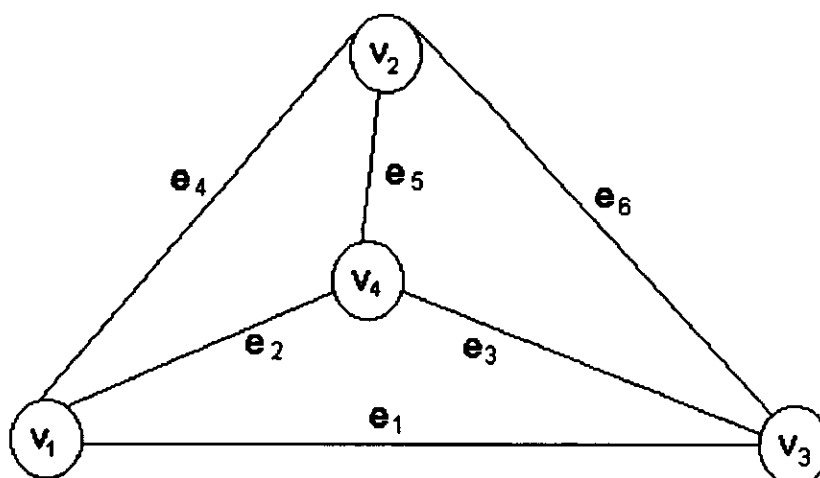


Figura B.1 Gráfica G de las relaciones entre los ejes estructurales.

Las áreas de síntesis serán las marcadas con d₁ a d₄.

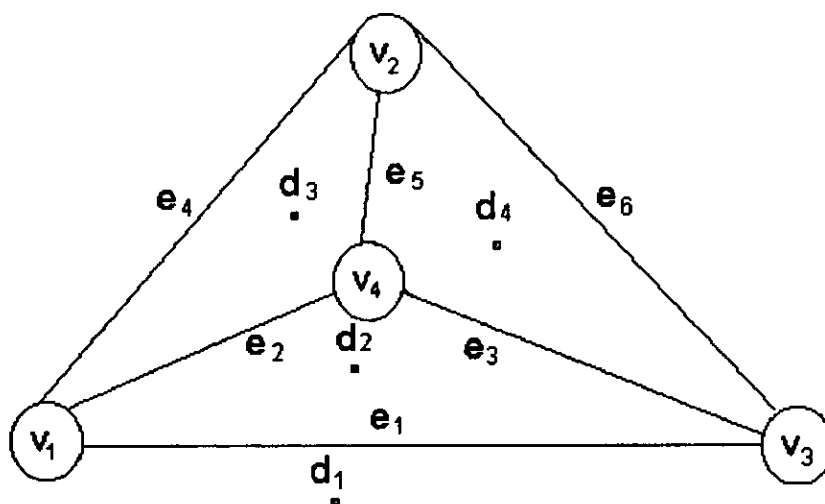


Figura B.2 Gráfica G con las Áreas De Síntesis d.

La representación matricial de las relaciones anteriores (Matriz de Adyacencias) es la siguiente:

	Trinomio E-SP-E	Curriculum	Docencia	Cultura Org. Ac.	Grados
	V ₁	V ₂	V ₃	V ₄	
V ₁	0	1	1	1	3
V ₂	1	0	1	1	3
V ₃	1	1	0	1	3
V ₄	1	1	1	0	3

$$\Sigma g = 12$$

$$\text{Nº de ramas} = 6$$

$$\text{rango } r = n - 1 = 4 - 1 = 3$$

$$\text{nulidad } M = e - (n - 1) = 6 - 3 = 3$$

1)

Matriz de Adyacencias de la gráfica G.

1 Indica cierta relación entre dos vértices.

0 Indica relación nula entre dos vértices.

Determinación de la gráfica dual G_d .

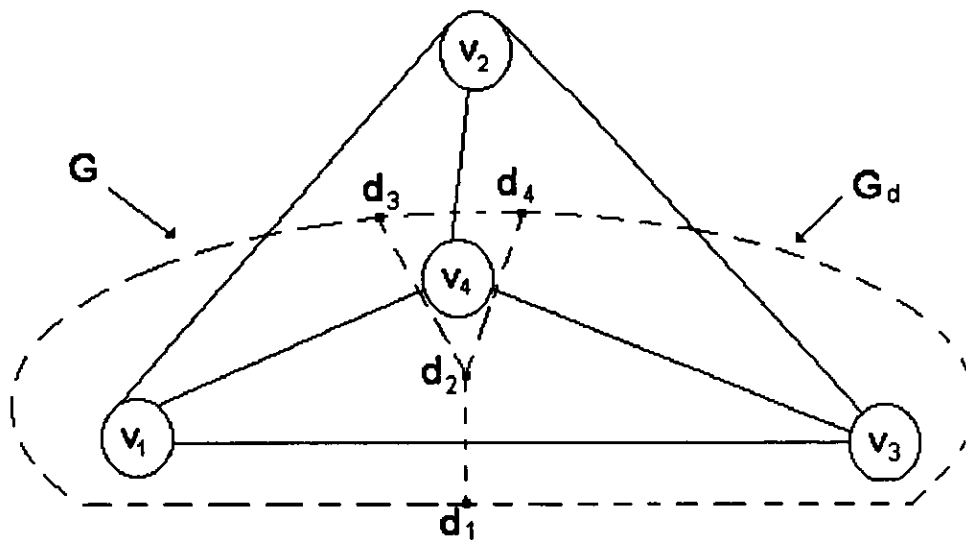


Figura B.3 Gráfica dual G_d y la gráfica primal G .

d_1, d_2, d_3, d_4 son las regiones determinadas por G , si relacionamos estas regiones obtenemos la gráfica dual G_d .

La representación matricial de la gráfica dual G_d es como sigue:

	d_1	d_2	d_3	d_4	Grado
d_1	0	1	0	0	1
d_2	1	0	1	1	3
d_3	0	1	1	1	3
d_4	0	1	1	0	2

$$\sum g = 9$$

$$\text{Nº de ramas } e_d = 6$$

$$\text{rango } r_d = n - 1 = 4 - 1 = 3$$

$$\text{nulidad } M_d = e_d - (n - 1) = 3$$

Nótese que:

$$e = e_d = 6.$$

$$r = r_d = 3.$$

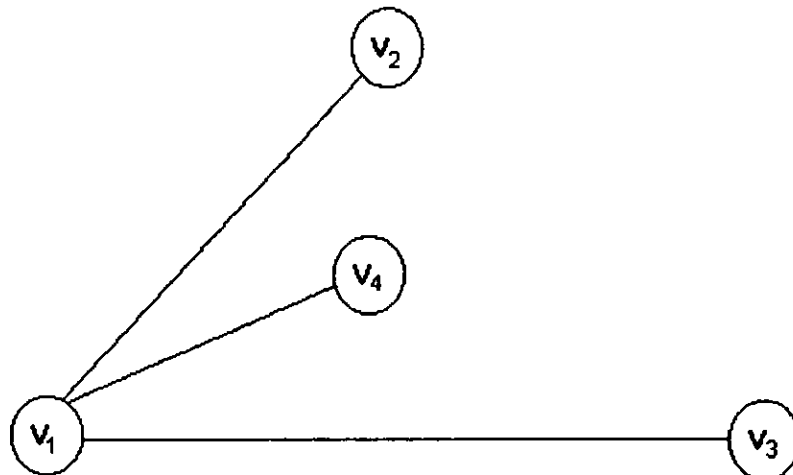
$$M = M_d = 1.$$

Análisis de Áreas de Síntesis fundamentales.

Determinación de los Árboles de Expansión de las gráficas primal y dual de la Figura B.3.

Para cada gráfica G y G_d tenemos 16 Árboles de Expansión (n^{n-2}).

Para la gráfica primal G consideramos el Arbol de Expansión que nos da las relaciones más relevantes.



$$T = \{ 1, 1, 0, 1, 0, 0 \}$$

Figura B.4 Árbol de Expansión T de la Gráfica Primal.

Para la Gráfica G_d de los 16 Árboles disponibles, podemos considerar, para fines de exponer la metodología, que dos de ellos tienen relaciones importantes, el T_{d1} y el T_{d2} .

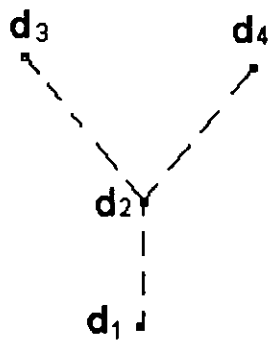


Figura B.5 Árbol de Expansión de la gráfica G_d .

Otra alternativa para el Árbol de Expansión de G la Gráfica Dual G_d .

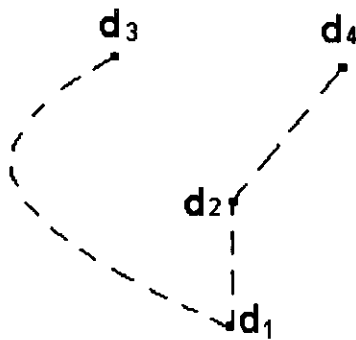


Figura B.6 Árbol de Expansión de T_{d2} de la Gráfica dual G_d .

Árboles de Expansión T y T_d de las Gráficas primal y dual (Figura 7.3).

Relación entre el Árbol de Expansión T y los árboles T_{d1} y T_{d2} .

Para el Arbol de Expansión T_{d1} .

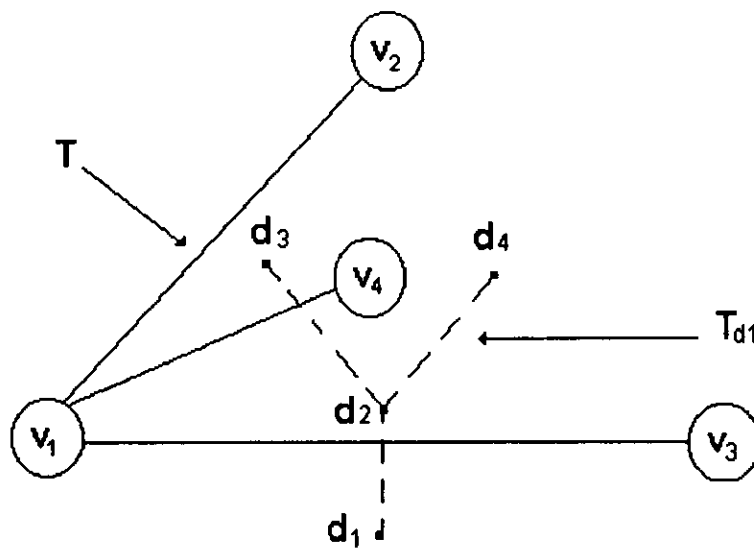


Figura B.7 Relación entre el Árbol de Expansión T de la Gráfica primal G y el Árbol de Expansión T_{d1} de la gráfica Dual G_d .

Para el Árbol de Expansión T_{d2} .

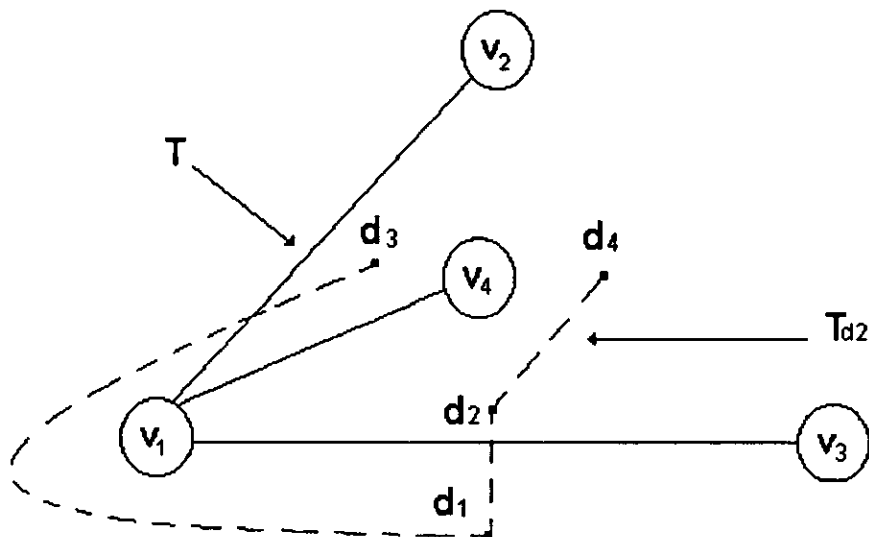


Figura B.8 Relación entre el Árbol de Expansión T de la Gráfica Original G y el Árbol de Expansión T_{d2} de la Gráfica Dual G_d .

Las ramas del Árbol de Expansión de la Gráfica Dual G_d que cruzan a las ramas del Árbol de Expansión de la Gráfica Primal G definen las Adyacencias fuertes entre esas áreas, lo cual nos indica las relaciones originales en G que se deberán reforzar.

Las relaciones posibles a reforzar en G son:

Para el Árbol T_{d1} :

$e_1 = (v_1, v_3)$: La docencia ante la formación y capacitación para y en el trabajo.

$e_2 = (v_1, v_4)$: Reestructuración de la normatividad congruente a la flexibilidad de formación y capacitación.

Para el Árbol T_{d2} :

$e_1 = (v_1, v_3)$: La docencia ante la formación y capacitación para y en el trabajo.

$e_4 = (v_1, v_4)$: Pertinencia del currículum ante las necesidades del sector productivo.

Para elegir el Árbol de la Gráfica dual, T_{d1} o T_{d2} , analicemos las diferencias en las relaciones a reforzar en G y tomemos las que sean más relevantes.

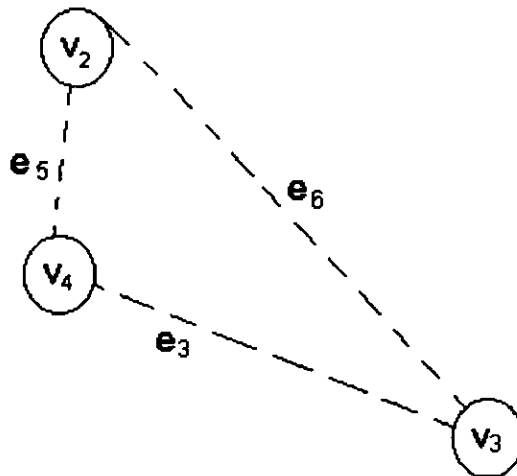
Diferencias en las relaciones a reforzar en G :

Para T_{d1} e_2

Para T_{d2} e_4

Consideramos que la relación más relevante es e_2 , por lo tanto el Árbol de Expansión con el cual trabajaremos será T_{d1} .

Análisis de las cuerdas del Árbol T y circuitos básicos.



$$M = G - T = (0, 0, 1, 0, 1, 1)$$

$$M = (e_3, e_5, e_6)$$

Figura B.9 Subgráfica M de cuerdas referentes al árbol de expansión T.

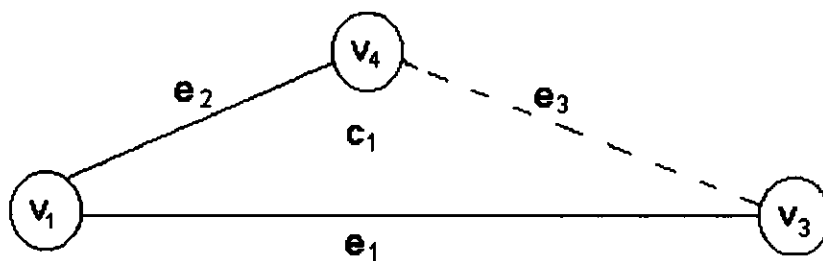


Figura B.10 Circuito Básico $c_1 = (e_1, e_2, e_3)$

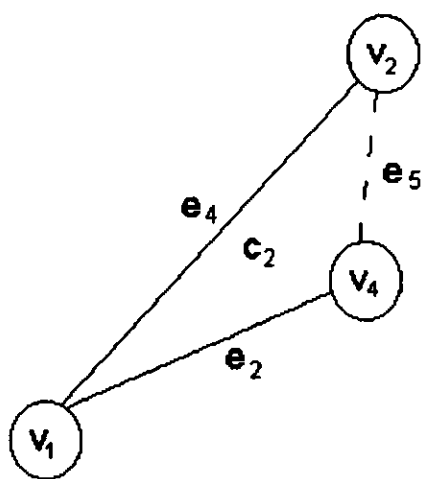


Figura B.11 Circuito Básico $c_2 = (e_2, e_4, e_5)$

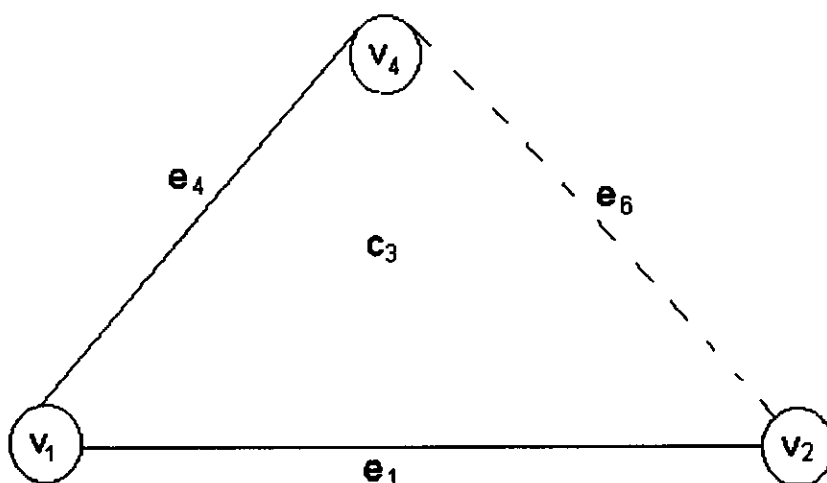


Figura B.12 Circuito básico $C_3 = (e_1, e_4, e_6)$

Matricialmente los circuitos fundamentales los podemos determinar como sigue:

$$C^+ = \begin{array}{c} \begin{array}{c} e_3 \quad e_5 \quad e_6 \\ \hline \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \end{array} \quad \begin{array}{c} e_1 \quad e_2 \quad e_4 \\ \hline \begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{array} \end{array} \\ C^+ = \begin{array}{c} \begin{array}{c} \\ \hline \end{array} \quad I \end{array} \quad \begin{array}{c} C^- \begin{array}{c} \\ \hline \end{array} \end{array} \end{array}$$

Los circuitos fundamentales son:

$$C_1 = \{ 1, 1, \underline{1}, 0, 0, 0 \}$$

$$C_1 = \{ e_1, e_2, \underline{e_3} \}$$

$$C_2 = \{ 0, 1, 0, 1, \underline{1}, 0 \}$$

$$C_2 = \{ e_1, e_2, \underline{e_5} \}$$

$$C_3 = \{ 1, 0, 0, 1, 0, \underline{1} \}$$

$$C_3 = \{ e_1, e_4, \underline{e_6} \}$$

Matriz de Cortes fundamentales.

Los cortes fundamentales son aquellos que cortan las cuerdas del árbol T.

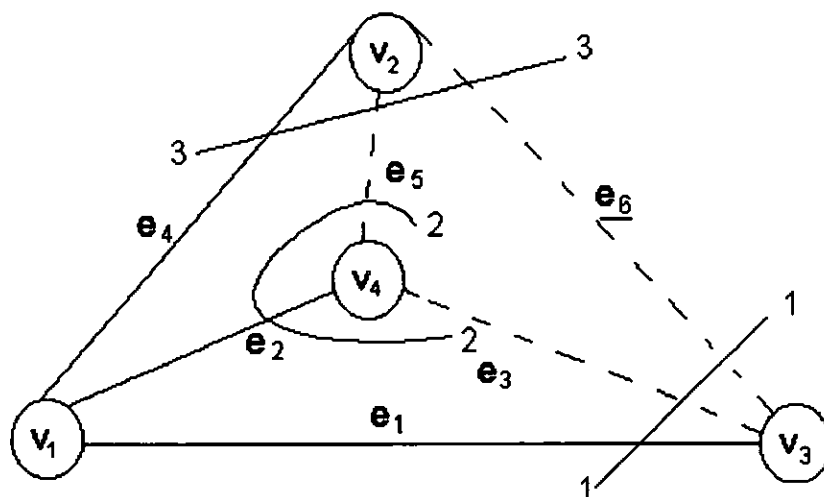


Figura B.13 Cortes fundamentales.

De la matriz de cortes fundamentales C^+ .

$$C^+ := \begin{array}{c|cc} & e_1 & e_2 & e_4 \\ \hline 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 \\ 3 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{c|c} C^- \\ \hline I \end{array} = \begin{array}{c|cc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

Transponiendo la matriz anterior obtenemos la Matriz de Cortes Fundamentales K.

$$K = \begin{vmatrix} C \\ I \end{vmatrix}^T$$

$$K = \begin{matrix} & \begin{matrix} e_3 & e_5 & e_6 & e_1 & e_2 & e_4 \end{matrix} \\ \begin{matrix} k_1 \\ k_2 \\ k_3 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

$$k_1 = \{ e_3, e_6, e_1 \}$$

$$k_2 = \{ e_3, e_5, e_2 \}$$

$$k_3 = \{ e_5, e_6, e_4 \}$$

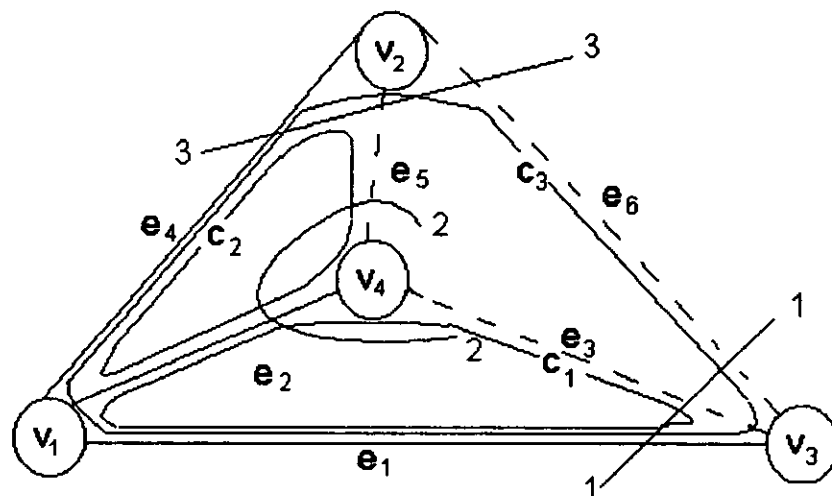


Figura B.14 Circuitos y cortes fundamentales.

Los circuitos y cortes fundamentales forman bases en los Subespacios Vectoriales de C_G y C_K del Espacio Vectorial E_G que define la gráfica G .

Cualquier circuito C_x o corte K_x se puede expresar como

$$C_x = a_1 C_1 + a_2 C_2 + a_3 C_3$$

y

$$K_x = a_1 k_1 + a_2 k_2 + a_3 k_3$$

Donde $a_1, a_2, a_3 \in \{0,1\}$

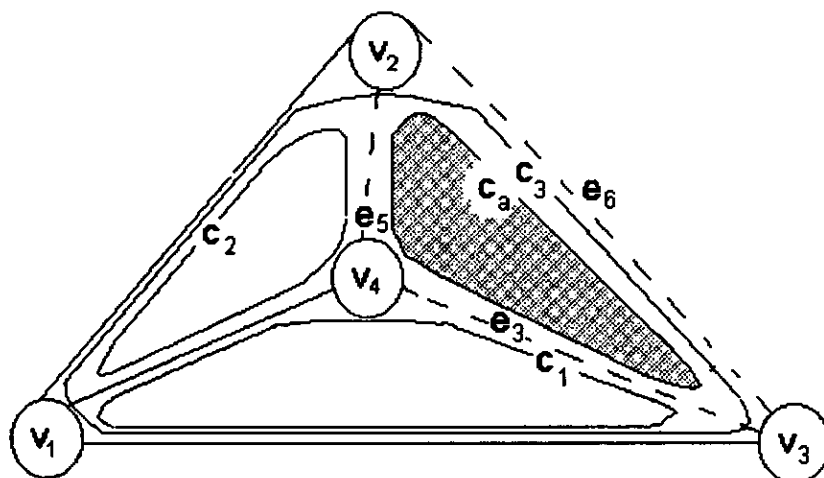


Figura B.15 Subgráfica del circuito C_a expresado como una combinación lineal de los circuitos básicos c_1, c_2, c_3 .

$$C_a = c_1 + c_2 + c_3$$

$$C_a = (e_1, e_2, e_3) + (e_2, e_4, e_5) + (e_1, e_4, e_6)$$

$$C_a = (1, 1, 1, 0, 0, 0) +$$

$$(0, 1, 0, 1, 1, 0) +$$

$$(1, 0, 0, 1, 0, 1)$$

$$C_a = (0, 0, 1, 0, 1, 1)$$

$$C_a = (e_3, e_5, e_6)$$

$e_1 = (v_1, v_3)$: La docencia ante la formación y capacitación para el trabajo.

$e_2 = (v_1, v_4)$: Reestructuración de la normatividad congruente a la flexibilidad de formación y capacitación.

$e_3 = (v_3, v_4)$: Congruencia de las condiciones interiores de trabajo para la versatilidad de la oferta educativa que brinda el MEBC.

- $e_4 = (v_1, v_2)$: Pertinencia del curriculum ante las necesidades del Sector Productivo.
- $e_5 = (v_2, v_4)$: Válidez oficial de los egresados de las diferentes modalidades educativas.
- $e_6 = (v_2, v_3)$: Formación (pedagógica) y actualización (disciplinaria) del docente para atender calidad a lo establecido en el currículum.
- e_1 : Docencia para el trabajo.
- e_2 : Normatividad congruente.
- e_3 : Congruencia laboral.
- e_4 : Pertinencia ante la producción.
- e_5 : Válidez oficial.
- e_6 : Actualización docente (Pedagógica y Disciplinaria).

RESUMEN DE LAS RELACIONES RELEVANTES, DE CIRCUITOS BÁSICOS Y CIRCUITOS INFERIDOS, ASÍ COMO ÁREAS DE SÍNTESIS Y SIGNIFICADOS.

Áreas de síntesis.

De las regiones conformadas en la gráfica tenemos:

Región d_1 .

Partimos de que el contexto general es una situación de crisis que solo se puede remontar aumentando por medio de una educación congruente con las necesidades de la producción.

Región d_2 .

En ésta área de síntesis concluimos que para tener una docencia que induzca directamente al trabajo es determinante tener una **normatividad congruente** en la institución educativa con respecto del reconocimiento de las competencias adquiridas y a su vez una **congruencia laboral** en el sentido de que el docente, no será un profesor tradicional ya que debe estar capacitado en el ámbito pedagógico y en su disciplina correspondiente a nivel teórico - práctico.

Región d3.

La pertenencia ante el sector productivo de la educación está determinada por las condiciones laborales de los docentes (Normatividad congruente) y el reconocimiento oficial de las competencias adquiridas por los egresados, en el campo laboral o en la escuela.

Región d4.

La actualización pedagógica y disciplinaria del docente implica una nueva reglamentación laboral, ya que esta por lo menos la capacitación disciplinaria por lo menos en buena parte deberá ser en los centros de producción y esta actividad debe ser reconocida oficialmente, es decir, debe existir una congruencia en sus condiciones laborales con las actividades.

1. Ramas del Árbol de Expansión.

$$1.1 \cdot e_1 = (v_1 , v_3)$$

1. Término: La docencia ante la formación y capacitación para el trabajo.

$$1.2 \cdot e_2 = (v_1 , v_4)$$

2. Término: Reestructuración de la normatividad congruente a la flexibilidad de formación y capacitación.

$$1.3 \cdot e_4 = (v_1 , v_2)$$

3. Término: Pertinencia del curriculum ante las necesidades del Sector Productivo.

2 Cuerdas del Árbol de Expansión.

$$2.1 \cdot e_3 = (v_1 , v_4)$$

6. Término: Congruencia de las condiciones interiores de trabajo para la versatilidad de la oferta educativa que brinda el MEBC.

$$2.2 \quad e_5 = (v_2, v_4)$$

7. Término: Válidez oficial de los egresados de las diferentes modalidades educativas.

$$2.3 \quad e_6 = (v_2, v_3)$$

8. Término: Formación (pedagógica) y actualización (disciplinaria) del docente para atender calidad a lo establecido en el curriculum.

3 Áreas de Síntesis.

$$3.1 \quad d_1 \rightarrow$$

13. Término: Efecto, en un contexto de crisis, de profesionales preparados a nivel licenciatura o técnico medio capacitados para adquirir competencias necesarias para la producción y desarrollo de la sociedad .

$$3.2 \quad d_2 \rightarrow \begin{matrix} (v_1, v_3, v_4) \\ \text{ó} \\ (e_1, e_2, e_3) \end{matrix}$$

14. Término: Cumplir con la función social y educativa de la institución ante las necesidades económicas y sociales del país.

$$3.3 \quad d_3 \rightarrow \begin{matrix} (v_1, v_2, v_4) \\ \text{ó} \\ (e_2, e_4, e_5) \end{matrix}$$

15. Término: La Cultura Organizacional Académica permite la pertinencia de los contenidos ofertados en sus diferentes modalidades que satisfacen las necesidades de desarrollo del país.

$$3.4 \quad d_4 \rightarrow \begin{matrix} (v_2, v_3, v_4) \\ \text{ó} \\ (e_3, e_5, e_6) \end{matrix}$$

16. Término: La Cultura Organizacional Académica permite que la docencia participe en el diseño curricular acorde a la demanda del sector productivo.

4 Circuitos fundamentales.

$$4.1 \quad C_1 = \begin{pmatrix} v_1, v_3, v_4 \\ \text{ó} \\ e_1, e_2, e_3 \end{pmatrix}$$

21. Término: Congruencia de las condiciones internas de trabajo con la versatilidad de la oferta educativa que brinda el MEBC.

$$4.2 \quad C_2 = \begin{pmatrix} v_1, v_2, v_4 \\ \text{ó} \\ e_2, e_3, e_5 \end{pmatrix}$$

22. Término: Validez oficial de los estudios realizados por los egresados de las diferentes modalidades educativas.

$$4.3 \quad C_3 = \begin{pmatrix} v_1, v_2, v_3 \\ \text{ó} \\ e_1, e_4, e_6 \end{pmatrix}$$

23. Término: Formación (pedagógica) y actualización disciplinaria permanente del docente para atender con calidad a lo establecido en el currículum.

Resumen tabular de los circuitos basicos y circuitos inferidos.

Rama en el árbol T	Cuerda del árbol T , relación inferida que encierra el circuito básico	Circuito Básico	Inferencia de síntesis en Circuitos Inferidos
$e_1 = (v_1, v_3)$	$e_3 = (v_3, v_4)$	$C_1 = \{ e_1, e_2, e_3 \}$	
	$e_6 = (v_2, v_3)$	$C_3 = \{ e_1, e_4, e_6 \}$	$C_a = \{ e_3, e_5, e_6 \}$
$e_2 = (v_1, v_4)$	$e_3 = (v_3, v_4)$	$C_1 = \{ e_1, e_2, e_3 \}$	
	$e_5 = (v_2, v_4)$	$C_2 = \{ e_2, e_4, e_5 \}$	
$e_4 = (v_1, v_2)$	$e_5 = (v_2, v_4)$	$C_2 = \{ e_2, e_4, e_5 \}$	$C_a = \{ e_3, e_5, e_6 \}$
	$e_6 = (v_2, v_3)$	$C_3 = \{ e_1, e_4, e_6 \}$	

BIBLIOGRAFIA.

BIBLIOGRAFIA

1. Proyecto experimental de formación para el trabajo basada en competencias.
Dirección General de Centros de Formación para el Trabajo;
Edi. SEP - SEIT;
México 1996.
2. Competencia Laboral y Educación Basada en Normas de Competencia.
Aut. Antonio Arguelles. / Compilador;
Edi. LIMUSA, SEP, CNCCL, CONALEP;
México 1996.
3. El camino sin sentido.
Tres ensayos sobre la Educación Técnica y la Formación Extra Escolar en México.
Aut. Jorge Munguía Espitia;
Edi. Universidad Pedagógica Nacional;
México 1995.
4. Reforma Estructural de la Formación y Capacitación de los Recursos Humanos en México.
Aut. Agustín E. Ibarra Aldama;
Edi. En la Memoria del Seminario Estrategia y Desarrollo del Proyecto de Modernización de la Educación Técnica y la Capacitación;
San Juan del Río Querétaro 1995.
5. Sistemas Educativos Basados en Competencias de Francia, Estados Unidos y Canadá.
Edi. IPN Secretaría Académica, Dirección de Educación Media Superior, División de Desarrollo Académico;
México 1996.
6. Modelo Curricular de Educación Basada en Competencias. Primer documento de trabajo.
Edi. IPN Secretaria Académica;
Dirección de estudios profesionales;
México 1995.

7. IPN Educación Basada en Competencias. Proyecto de Desarrollo en el nivel medio superior. Guía de operación.
Ed. Secretaría Académica. Dirección de Educación Media Superior.
División de Desarrollo Académico;
México 1997.
8. El Modelo Curricular de Educación Basada en Competencias. Posibilidades y retos para su implantación.
Revista Académica No. 1; Educación, Investigación, Vinculación Páginas 11-19.
Edi. IPN
Febrero de 1996.
9. Centro Experimental de Educación Basada en Competencias.
Centro de Estudios Tecnológicos No. 1.
M. en C. (c) Enrique R. Ojeda Barranco. Director del plantel.
Propuesta elaborada por:

M. en C. (c) Enrique R. Ojeda Barranco.
Lic. Beatriz Jasso Arizmendi.
Ing. Horacio Armenta Sosa.
M. en C. Manuel Montiel Escobar.
Lic. Francisco Vega Hernández.
Lic. Leticia Sánchez Vargas.
Ing. Arquitecto Isidoro Delgado Tovar.
Ing. J. Carlos V Peralta Guarneros.
M. en C. Cornelio Yañez Marques.
Ing. R. Jorge Ambrosio Hernandez.
Lic. Virginia Davalos Osorio.
10. Pearls in Graph Theory.
A Compresive Introduction.
Aut. Nora Hartsfield, Gerhard Ringel.
Ed. Academic Press, Inc. 1990.
11. Graph Algorithms.
Aut. Shimon Even.
Ed. Computer Science Press. 1979.
12. Aplicattions of Graph Theory Algorithms.
Aut. Vinod Chachra, Prabhakar M.
Ghare, James M. Moore.
Ed. Elsevier North Holland. Inc. 1979.

13. Introducción a la Combinatoria y sus aplicaciones.
Aut. A. Kaufmann.
Ed. C. E. C. S. A. 1971.
14. Exercices et Problèmes de Recherche opérationnelle.
Aut. Gérard Deezbazeille.
Ed. Dunod 1976.
15. Enfoque de Sistemas en la Educación, Teoría de Gráficas.
Aut. Dr. Javier Salazar Resines.
Ed. Limusa 1979.