



UNIVERSIDAD TÉCNICA NACIONAL
VICERRECTORÍA DE INVESTIGACIÓN Y POSTGRADO
CENTRO DE FORMACIÓN PEDAGÓGICA Y
TECNOLOGÍA EDUCATIVA
MAESTRÍA EN ENTORNOS VIRTUALES DE APRENDIZAJE

PROYECTO DE INTERVENCIÓN

TÍTULO DEL PROYECTO:
ESTRUCTURAS DISCRETAS PARA INFORMÁTICA
BAJO UN ENFOQUE BIMODAL

PREPARADO POR:
ENRIQUE VÍLCHEZ QUESADA

TUTOR DEL PROYECTO:
MARIELA DELAURO

AÑO 2018

ÍNDICE

RESUMEN TÉCNICO	4
PROPUESTA DEL PROYECTO	5
1. El problema	6
▪ Problema	9
▪ Justificación.....	9
▪ Contexto del problema	11
2. Prospectiva.....	14
3. Propuesta pedagógica.....	17
4. Objetivos	19
▪ General	19
▪ Específicos	19
5. Resultados esperados.....	19
6. Aspectos operativos.....	21
▪ Administración	21
▪ Aprendizaje y tecnologías.....	22
▪ Tutoría.....	29
▪ Materiales didácticos	30
7. Evaluación y seguimiento del proyecto	34
▪ Antes, durante y al finalizar el proyecto.....	34
▪ Indicadores de evaluación de cada aspecto operativo	36
i. Gestión del curso	36
ii. Materiales didácticos	37
8. Cronograma para la ejecución del proyecto	38
9. Presupuesto	42
10. Bibliografía.....	43
DESARROLLO DEL PROYECTO	45
1. Nombre del curso virtual	46
2. Selección y justificación de las herramientas tecnológicas.....	46
3. Planificación de las clases	54
4. Redacción de las clases	69
5. Captura de pantalla de las clases.....	87
DOCUMENTOS ELABORADOS	101

Guía didáctica.....	102
Módulo didáctico.....	134
Conclusiones.....	217

RESUMEN TÉCNICO

El presente trabajo describe el proceso de diseño y desarrollo de un entorno virtual de aprendizaje para la materia *EIF-203 Estructuras Discretas para Informática* que forma parte del plan de estudios de la carrera *Ingeniería en Sistemas de Información* de la Universidad Nacional de Costa Rica (UNA). Este entorno responde a una serie de necesidades pedagógicas sentidas en el contexto de la cátedra *EIF-203* reconociendo el impacto positivo que puede tener el uso de las tecnologías digitales dentro del marco de una planificación didáctica seria y en concordancia con el modelo pedagógico integrado de manera institucional en la UNA.

La propuesta posee una fundamentación que abarca el planteamiento de un problema y su justificación, una prospectiva, una respuesta pedagógica y aspectos operativos inmersos dentro de un proyecto educativo que traslada a la cátedra del curso *EIF-203* a un escenario donde se abandona la presencialidad como el único dominio de enseñanza y aprendizaje para dar paso a la virtualidad en una combinación enriquecedora de ambos espacios formativos.

La Universidad Nacional de Costa Rica desde su fundación ha sido tradicionalmente una institución universitaria de naturaleza presencial, por lo que el planteamiento de este proyecto constituye una innovación pedagógica disruptiva. Esta innovación encontró su desarrollo bajo una perspectiva pedagógica que reconoce el aprendizaje como un resultado social y activo, de construcción de conceptos y resolución de problemas, aproximando los contenidos de clase a situaciones reales simuladas por el uso de las tecnologías con fines educativos. Lo anterior, resignifica el ejercicio docente hacia experiencias de aprendizaje donde los educandos ocupen una posición central en la dinámica de una red de conexiones dentro de un ecosistema digital como nicho subyacente de la eclosión de un nuevo conocimiento y sus aplicaciones.

El entorno virtual de aprendizaje creado para el curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal, se ha implementado en el aula virtual institucional de la UNA, ofreciendo refrescantes alternativas de aprendizaje en las clases diseñadas.

PROPUESTA DEL PROYECTO

1. El problema

En el contexto de la Universidad Nacional de Costa Rica emergió en el año 2012 la iniciativa de incorporar el uso de las tecnologías relacionadas con el aprendizaje virtual y el asistido por ordenador en distintos cursos ofrecidos por la Escuela de Informática (EI). La Escuela de Informática es una unidad académica conformada por diversos profesionales que imparten cursos en cuatro áreas de especialidad: ingeniería de software, ingeniería de sistemas de información, arquitectura y telecomunicaciones y, formación complementaria (humanística, matemática, idiomas, economía y administración).

La Escuela de Informática de la UNA, ante los retos impuestos por el uso de las tecnologías y sus bondades y limitaciones en el ámbito educativo, ha promovido varios proyectos de investigación en docencia relacionados con su uso sistemático. En un inicio, dentro de un marco institucional que desde el año 2012 ha impulsado una política de desarrollo donde las tecnologías ocupan un lugar preponderante en la pedagogía, los proyectos se han centrado en la capacitación de los docentes hacia el uso tecnológico de ciertas herramientas, quedando relegada la formación en la conceptualización y diseño de cursos en entornos de aprendizaje virtual.

En el año 2012, en el marco del proyecto de investigación denominado: “*Facebook como herramienta de enseñanza y aprendizaje para el curso EIF-203 Estructuras Discretas para Informática a través del uso de cuadernos interactivos*”, la iniciativa contempló en un período de dos años las etapas de planificación general, operativa, ejecución y evaluación del curso *EIF-203 Estructuras Discretas para Informática* con una modalidad que combinó la presenciabilidad con la virtualidad, lo cual resultó toda una innovación pedagógica dentro de la carrera *Ingeniería en Sistemas de Información* de la EI de la UNA, por dos razones principales: en primer lugar los docentes y estudiantes en su mayoría no tenían formación académica y experiencias previas en el aprendizaje virtual como otra modalidad de estudio y en segundo lugar, en la Escuela de Informática ya se hacía necesario un curso que capacitara y formara a los estudiantes en el uso de ciertas tecnologías con fines didácticos y de investigación, además del empleo de una

plataforma de teleformación, ¡un gran reto, si se piensa en las características de una tradición pedagógica dentro de una universidad con una fuerte tendencia hacia lo presencial y el escaso uso de tecnologías para la enseñanza (Vílchez, 2014). Posiblemente dentro de otros contextos el proyecto no hubiera resultado tan innovador como ocurrió en la UNA.

Es importante reconocer que los esfuerzos realizados en la *EI* de la Universidad Nacional de Costa Rica parten de un pensamiento docente colectivo que mira hacia el futuro, que pretende construir para el futuro y que reconoce su responsabilidad al respecto como un grupo encargado de formar y sensibilizar a las nuevas generaciones interesadas en el sector informático y tecnológico de nuestro país.

Pese a ello, las etapas de un proceso de planificación serio pecan por algunos de los inconvenientes suscitados en las transformaciones organizacionales que apuntan al cambio, muchas veces con poco tiempo de análisis ante sus limitaciones y posibilidades reales.

Los procesos de diagnóstico realizados en la Universidad Nacional de Costa Rica no han sido lo suficientemente estructurados como para prever las necesidades de infraestructura y de equipos informáticos actualizados al servicio del cuerpo docente y las grandes cantidades de estudiantes que provienen de zonas rurales del país y que no cuentan con un computador en sus residencias temporales. Esto desde luego, está ocasionando graves problemas de acceso a los alumnos y los profesores. Los estudiantes se quejan constantemente de sus limitaciones económicas y pocas facilidades para realizar sus tareas virtuales y muchos docentes se disputan semana a semana el uso de los recursos disponibles, por ejemplo, el cañón de proyección o el único laboratorio de informática con el que cuenta su unidad, para atender la demanda académica semestre a semestre.

La prospectiva (Prieto, 2008a) en la Escuela de Informática de la UNA tiene un comportamiento disgregado, algunos docentes cercanos a las políticas de la unidad,

apoyan todos los procesos de innovación pedagógica a través del uso de las tecnologías y otros se resisten a su uso. Aunque este punto no preocupa tanto, pues en el ámbito educativo el convencimiento profesional es una tarea bipartita entre el tiempo y la capacitación docente.

La gestión académica de la unidad está derivando hacia una planificación general cuyos resultados convergen en la articulación de distintos proyectos de investigación, incluidos en áreas estratégicas de intereses macros (UNA) y micros (de la *EI* propiamente). La evaluación de todos estos procesos se está realizando de forma desintegrada, a futuro inmediato es un aspecto fundamental por considerar dentro de la gestión general.

En el año 2016 iniciamos el reto de incorporar dentro de la formación de futuros ingenieros en informática, un curso donde los alumnos logaran comprender y aplicar el uso apropiado de distintos tipos de tecnologías para la investigación y resolución de problemas en el campo de la matemática discreta (Vílchez, 2017).

Lo interesante de esta experiencia ha consistido en el ambiente institucional desde donde la idea emergió, en una universidad que por décadas ha sido predominantemente presencial y la docencia universitaria ha demostrado un claro énfasis en la experticia por los contenidos y no tanto en las prácticas pedagógicas de sus profesores. Este último aspecto ha llamado la atención de las autoridades universitarias actuales, proponiendo formalmente en su gestión un modelo pedagógico definido y reconocido organizacionalmente.

En este sentido, el curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal, se propuso con el objetivo de iniciar el desarrollo de las competencias requeridas por los estudiantes en la aventura del empleo experimental de recursos didácticos basados en tecnologías digitales tales como Internet, un aula virtual y el uso de software especializado, todo ello, con fines educativos. El plan inició en el año 2012 y sus resultados han sido variables por razones que se circunscriben en problemas

clasificados en las dimensiones: pedagógica, tecnológica, de tradición institucional y de las características del sistema educativo nacional.

- Problema

Justamente en el contexto expuesto, surge un problema pragmático muy claro; continuar con el curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal o volver a impartir la materia con sus características netamente presenciales.

El problema trasluce elementos de mayor complejidad, creemos y demostramos en el año 2013 (Vílchez y González, 2014) que el curso *EIF-203* fortalece las competencias de los alumnos en cuanto al uso de recursos didácticos innovadores, además de mejorar sus condicionamientos cognitivos generando oportunidades para favorecer el autoaprendizaje y el coaprendizaje. Sin embargo, ¿por qué en años posteriores los resultados no han sido, en algunos casos, los esperados? Tenemos algunas hipótesis y este es el problema que debemos resolver. Consideramos que el curso *EIF-203 Estructuras Discretas para Informática* ofrece verdaderamente los espacios de enseñanza y aprendizaje descritos, pero para ello, el perfil docente, la propuesta pedagógica, los insumos tecnológicos ofrecidos por la institución a profesores y estudiantes, la normativa que rige la implementación de estas modalidades, la resistencia al cambio y la herencia cultural de un sistema educativo predominantemente presencial y positivista, requieren de un análisis de construcción de futuro que albergue los retos y alternativas de solución enmarcados no solo en la coordinación de la materia, sino también, en la unidad académica en su conjunto y en la universidad en general.

- Justificación

Construir futuro en la Escuela de Informática de la Universidad Nacional de Costa Rica, no significa ir en concordancia ciega con las nuevas o relativamente nuevas tendencias hacia el uso de las tecnologías digitales, significa pensar y repensar la manera en cómo ellas pueden impactar el futuro de la educación para las ingenierías.

Buscamos soluciones en el mediano y largo plazo que tengan una repercusión en las prácticas educativas de los docentes y por tanto en las prácticas de aprendizaje de los estudiantes. Con frecuencia llamamos a nuestros actuales educandos “nativos digitales”, sin embargo, qué significa realmente ser un nativo digital, ¿bastará con haber crecido presionando botones y conviviendo con pantallas?, pareciera que esta es la convicción irracional de muchos. Ciertamente en esta época confusa nos encontramos con “cafres digitales” e “ilustrados digitales”, ¿qué estamos formando?, somos conscientes de lo que queremos y la dirección contextual que deben llevar las tecnologías hacia la estructuración de una sociedad conformada por personas competitivas, críticas y altruistas. Creemos que los pasos de transformación de esta utopía a un escenario real, inicia con acciones concretas, aventureras, pero no descabelladas, ambiciosas, pero no descontextualizadas, en el marco de lo idealizable, pero no de lo desproporcionado. A esto precisamente, aspira responder el proyecto del curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal, hay razones muy puntuales ya mencionadas en el apartado anterior, para justificar la continuidad de este trabajo dada su importancia y aportes institucionales.

En este año 2018 ante las dificultades presentadas, las limitaciones detectadas y la necesidad de no desperdiciar los recursos económicos y humanos de la UNA, es fundamental hacer un alto en el camino, no para negar la responsabilidad ya adquirida, sino para establecer un replanteamiento de esta cátedra, en cuanto a la estructura curricular del curso, lo que podemos actualmente lograr con las condiciones institucionales, la necesidad de crear un perfil docente y a partir de él la capacitación que posibilite a los profesores de la unidad académica dar un salto seguro y firme que consolide el curso con sus características virtuales. Nos encontramos ante un océano de oportunidades que con el tiempo creemos convergerán evolutivamente a una oferta de servicios, en temas relacionados con la calidad en la docencia semipresencial en Costa Rica.

- Contexto del problema

La Universidad Nacional constituye una de las cinco universidades públicas de Costa Rica, con una oferta educativa basada en la presenciabilidad desde su fundación. En la actualidad el modelo pedagógico en el que se sustenta se fundamenta en los siguientes principios:

- “Respeto a la diversidad en todas sus expresiones.
- Respeto y compromiso con la igualdad de oportunidades y con la construcción de una sociedad más justa y equitativa.
- Formación de profesionales solidarios y comprometidos con el bienestar social.
- Flexibilidad para conceptuar el aprendizaje como proceso sociocultural, histórico, dinámico y transformable, posible y que puede construirse de muchas maneras.
- Interacción en los procesos formativos donde los conocimientos sean discutidos y enriquecidos permanentemente.
- Formación de un espíritu investigador en los futuros profesionales.
- Creatividad que permita la innovación, así como la utilización de medios, estrategias y recursos de enseñanza en los procesos de mediación pedagógica.
- Disposición para determinar los principios lógicos subyacentes en cada disciplina, que permitan una formación profesional de calidad.
- Evaluación como proceso integral, concertado, permanente, contextualizado y propositivo.
- Mejoramiento continuo en la formación integral de los estudiantes y los procesos de gestión académica-administrativa y paraacadémica.
- Visión prospectiva que permite la planificación estratégica para el logro de objetivos a mediano y largo plazo.”

Del modelo principalmente es rescatable el compromiso organizacional que asumen todos los docentes hacia una cultura de planificación prospectiva, que permita a partir de la adaptación de sus estrategias de enseñanza y aprendizaje, alcanzar logros que a mediano y largo plazo posibiliten la innovación, el enriquecimiento de los contenidos y la utilización de medios y recursos creativos en la mediación pedagógica ¡La institución

reclama cambios!, invita a modernizar de manera contextual las prácticas educativas cotidianas sea en el ámbito presencial o semipresencial.

En particular la Escuela de Informática de la UNA, en consistencia con el modelo pedagógico anterior, se ha propuesto los siguientes objetivos:

- “Formar profesionales capaces de potenciar el uso de la información y de las herramientas tecnológicas de la computación y la informática como motor de desarrollo, calidad de vida y bienestar social.
- Contribuir al desarrollo de la sociedad costarricense formando profesionales con capacidad de liderazgo en la búsqueda de soluciones responsables y éticas a las demandas que plantea el desarrollo de proyectos de sistemas de información.
- Contribuir al desarrollo integral, autónomo y sostenible de la sociedad costarricense, formando profesionales críticos, reflexivos, independientes, participativos y creativos con dominio en el área de la ingeniería en informática.
- Preparar profesionales con capacidad de aprendizaje autodidacta y promover en ellos la autonomía intelectual.
- Formar profesionales con sentido de pertenencia cultural y valores que incluyen la tolerancia, la equidad, el respeto por las diferencias, y por la vida en todas sus formas.”

La unidad académica, en este sentido, da un especial énfasis a la vinculación con otras áreas de conocimiento, específicamente al área de las tecnologías digitales en cuanto al impacto que producen en el sector productivo, con sus limitaciones y sus fortalezas. A este respecto, el curso *EIF-203 Estructuras Discretas para Informática* representa bajo su modalidad semipresencial, una respuesta ante los retos impuestos no solamente por la nueva generación de jóvenes que ingresan a la UNA y a carreras relacionadas con la ingeniería, sino también, en el seno de una institución que tanto de forma macro como micro reconoce el valor y la importancia en la formación a través del uso didáctico de los distintos recursos disponibles de mediación pedagógica en correspondencia crítica con las nuevas tendencias.

La UNA además, el 30 de agosto del año 2008 aprobó el documento denominado: “Políticas para la incorporación de las tecnologías de información y la comunicación (TIC) en los procesos académicos de la Universidad Nacional” que para bien o para mal está regulando la incorporación de las tecnologías de información y comunicación en los procesos académicos institucionales, por medio de una entidad centralizada llamada: *UNA Virtual*, quien es la responsable de asesorar, promover y desarrollar iniciativas de incorporación de las TIC en el quehacer académico, además de administrar y desarrollar el aula virtual de toda la universidad implementada de manera oficial en la plataforma *Moodle*. Los resultados no han sido los mejores pues la centralización de servicios con una alta demanda ha ocasionado ruptura comunicacional, lentitud en la resolución de problemas, lentitud en la apertura y seguimiento de los cursos y poco codesarrollo de la plataforma *Moodle* a las necesidades particulares de la UNA y sus distintas Escuelas. Por otra parte, vislumbramos a futuro con mucho temor, una burocratización de los procedimientos para la innovación, pues si una unidad académica decide utilizar cierto tipo de TIC en sus procesos educativos, ha perdido parte de su hegemonía y libertad académica teniendo la obligación de hacer consulta primero al departamento *UNA Virtual*, ajeno y en muchos casos desconocedor de las necesidades particulares ¡Bien o mal este es nuestro marco jurídico!

La disponibilidad de laboratorios y licenciamiento de software es otro de los aspectos que preocupan a nivel organizacional. Ya existe un grupo de académicos de distintas Escuelas que están promoviendo el uso de software libre como solución ante las problemáticas emanadas por el software privativo y con una fuerte influencia provocada por la tendencia mundial que ha traído este movimiento a muchos países, como, por ejemplo: España. Pese a los esfuerzos aún estamos en camino y en una edad temprana. En la Escuela de Informática de la UNA, además, los recursos no son suficientes, tanto los audiovisuales como de acceso a los laboratorios disponibles para todos los cursos que se imparten semestre a semestre, un detalle operativo muy importante por las exigencias que demanda una modalidad híbrida.

Como parte de este ejercicio de diagnóstico, la caracterización de los alumnos que ingresan a las carreras relacionadas con ingeniería en la Universidad Nacional de Costa Rica se resume de manera genérica en los siguientes puntos:

- Muchos provienen de zonas rurales en las cuáles durante su educación primaria y secundaria tuvieron poco contacto con el uso de tecnologías digitales, inclusive, muchos entran a la universidad sin saber cómo crear siquiera una cuenta de correo electrónico.
- El curso *EIF-203 Estructuras Discretas para Informática* se encuentra en el segundo año de la carrera de *Ingeniería en Sistemas de Información*, por tanto, los estudiantes se muestran relativamente neófitos en actividades relacionadas con la lectoescritura y hábitos de investigación.
- Los alumnos prefieren en su mayoría las clases presenciales y magistrales, presentando una fuerte resistencia a la modalidad virtual.
- Los estudiantes no saben trabajar de manera colaborativa, ni cooperativa, aunque están acostumbrados a relacionarse de forma grupal, dividiéndose tareas y unificando regularmente sus resultados.

¡Y bien!, con este panorama general descrito, iniciaremos un análisis de un escenario que hemos idealizado desde hace varios años con la implementación del curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal y realizaremos un ejercicio retrospectivo contrastando la experiencia acumulada hasta la fecha y las características institucionales.

2. Prospectiva

La cátedra de la materia *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal, emergió con la idea de institucionalizar un curso de pregrado dentro de la formación de futuros ingenieros en informática, en una modalidad *blended-learning* que lograra aproximar a los estudiantes a otras realidades educativas concordantes con el uso de las tecnologías de la información y comunicación. Bajo este supuesto, el escenario ideal que orientó nuestras intenciones académicas se circunscribe en los siguientes aspectos:

- En la dimensión pedagógica: los docentes presentan una formación y experiencia adecuada en el diseño y manejo de un entorno de aprendizaje virtual, además son conscientes de las diferencias y similitudes entre los espacios educativos presenciales y virtuales. Se sienten comprometidos con una modalidad educativa bimodal, reconociendo el aporte que esta puede proporcionar en cuanto al desarrollo de competencias que, sin el uso de esta modalidad, sería más difícil integrar en la formación de los estudiantes. Los docentes presentan las habilidades necesarias para la elaboración de material didáctico dirigido a este tipo de escenarios educativos. Finalmente, los profesores tienen conocimiento sobre estándares de calidad en la educación a distancia.
- En la dimensión tecnológica: los profesores utilizan la tecnología con una posición pedagógicamente crítica, son usuarios, pero también codesarrollares, definiendo con claridad sus necesidades y los aspectos en los cuáles la plataforma *Moodle* de la UNA debería converger con ayuda del grupo *UNA Virtual*. Los docentes tienen formación en el uso de las tecnologías digitales. Poseen conocimiento de experiencias internacionales sobre el uso de estas tecnologías en el sector educativo. Son usuarios activos de la Red y las tecnologías *Web 2.0*, reconociendo desde su visión de consumo sus ventajas y limitaciones.
- En la tradición institucional: la Universidad Nacional de Costa Rica, reconoce la importancia y necesidad de crear espacios educativos semipresenciales y virtuales, funda un *Centro de Materiales Audiovisuales* que apoyan los procesos de enseñanza y aprendizaje a distancia. En su normativa institucional, se brinda una atención particular a las modalidades semipresenciales y virtuales, en cuanto al reglamento de evaluación de los aprendizajes y a los derechos y deberes tanto de los académicos como de los estudiantes. Institucionalmente se brinda a los distintos grupos interdisciplinarios el tiempo necesario dentro de su jornada, para la conceptualización, diseño e implementación de cursos y programas bajo estas modalidades.
- En el sistema educativo nacional: el *Ministerio de Educación Pública* con apoyo de los académicos de la Escuela de Informática de la Universidad Nacional, inicia procesos de formación y capacitación docente en tutoría virtual, entornos de

aprendizaje virtuales y capacitación específica en didáctica de las ingenierías, pedagogía y andragogía.

Por otra parte, vislumbrando el escenario posible en cada una de estas dimensiones, pretendemos en un plazo de tres años obtener los siguientes logros:

- En la dimensión pedagógica: docentes capacitados en el área de la tutoría virtual, al menos con un curso aprobado por una organización internacional. Hay claridad sobre el perfil de los profesores que constituyen la cátedra, lo cual proporcionará insumos importantes de contratación de personal. Se consolidará un grupo de tres o cuatro investigadores en el área para determinar en el contexto de la UNA de Costa Rica ventajas, desventajas y necesidades ante las exigencias de una modalidad *blended-learning*. Se contará con un grupo de tres a cuatro profesionales que, de manera permanente, se dedicarán a elaborar y reelaborar materiales didácticos para el curso *EIF-203 Estructuras Discretas para Informática*.
- En la dimensión tecnológica: los docentes que formarán parte de la cátedra se capacitarán en el uso de distintos recursos tecnológicos, recibirán para estos efectos el estímulo de contar con una beca semestral para participar de un curso en el Instituto de Capacitación y Asesoría Informática (ICAI), o bien, de cualquier otro centro de estudio de la Universidad Nacional de Costa Rica. Anualmente la cátedra del curso recogerá información para detectar las debilidades y fortalezas de la plataforma *Moodle* implementada en la UNA, esta información retroalimentará los esfuerzos del departamento *UNA Virtual* para robustecer y adecuar la plataforma institucional. Finalmente, un grupo de dos o tres especialistas actualizará a la cátedra con experiencias internacionales que alimenten el quehacer académico en el contexto de la materia *EIF-203 Estructuras Discretas para Informática*.
- En la tradición institucional: se solicitará a la dirección de la unidad académica un pronunciamiento anual si es necesario, de parte del Consejo de Unidad y Consejo de Facultad a la rectoría, para regularizar en el marco jurídico los derechos y

obligaciones de académicos y estudiantes en el contexto de una modalidad semipresencial y/o virtual.

- En el sistema educativo nacional: la Escuela de Informática ofrecerá prestación de servicios a la comunidad costarricense en general, para la implementación exitosa de cursos bimodales o virtuales. El proceso iniciará sin ningún costo ofreciendo asesoría al *Ministerio de Educación Pública* en temas relacionados con la tutoría virtual y la complejidad de los entornos de aprendizaje virtual.

La propuesta pedagógica de curso y sus objetivos bajo la modalidad *blended-learning*, se presentan en los siguientes apartados.

3. Propuesta pedagógica

En la cátedra del curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal, el uso, construcción y aplicaciones en el ámbito educativo de diversos recursos didácticos digitales, se fundamenta en el reconocimiento del aprendizaje como un producto social (Vygotsky, citado por Espiro, 2008a) y como un proceso activo (Ausubel y Bruner, citados por Espiro, 2008a) a partir del cual los alumnos construyen o reconstruyen ideas, conceptos, soluciones, estrategias, conjeturas y aplicaciones de la teoría a situaciones reales o hipotéticas en los diversos contextos que caracterizan al sistema productivo nacional en el campo de las tecnologías de la información.

En este curso se fomenta la construcción social del conocimiento mediante el trabajo en equipo, cooperativo y colaborativo. Se espera que los alumnos se apropien de habilidades y destrezas de investigación, que les permita tener una mayor independencia sobre sus propias estructuras de pensamiento, aprendizaje de estrategias de aprendizaje (aprender a aprender) y el desarrollo de un pensamiento crítico a través del ejercicio dialógico, el respeto por los saberes individuales y colectivos, y la creación conjunta de nuevas ideas, tal y como nos lo plantea la educación liberadora propuesta por Paulo Freire (Espiro, 2008b).

El andamiaje citado por Bruner y el aprendizaje significativo desarrollado por Ausubel, en muchas de las estrategias didácticas que caracterizan a este curso, cobran una posición central tanto en el ejercicio de la tutoría virtual y presencial, como en las actitudes que se espera desarrollar en los educandos. El docente asumirá el reto de elaborar y seleccionar materiales de apoyo (lecturas, presentaciones, *links*, recursos de la *Web 2.0*, problemas, entre otros) que inciten el descubrimiento de técnicas y métodos donde el alumno encuentre los espacios necesarios para una construcción y aplicación personal y colectiva de los conocimientos relacionados con el campo de la matemática discreta.

Esta propuesta pedagógica toma en consideración también, la importancia del aprendizaje receptivo y el condicionamiento operante propuesto por Skinner (Espiro, 2008a), en el ejercicio de la tutoría virtual estimulando la práctica de una conducta social adecuada y los fundamentos de una comunicación asertiva, respetuosa y tolerante, utilizando las distintas herramientas de comunicación tanto sincrónicas como asincrónicas del aula virtual. Además, se reconoce el valor del aprendizaje memorístico como puente para conciliar los conocimientos teóricos previos, con estrategias de enseñanza no mecanicistas.

Finalmente, se reconoce en el conectivismo la teoría de aprendizaje que permite comprender las interacciones sujeto-sujeto, sujeto-red, sujeto-artefacto, artefacto-artefacto, sujeto-comunidad, comunidad-comunidad implicadas en las nuevas formas de interacción e integración cognitiva que emergen como un producto de la expansión de los ecosistemas digitales (en mucho favorecida por la Internet) caracterizados por nuevas estructuras de comunicación, interrelación y operacionalización de procesos en la adquisición de conocimientos y génesis de nuevo contenido (Siemens, 2004). Los entornos de aprendizaje actuales fundamentados en una dinámica sociocultural globalizada no pueden ser explicados intentando llenar los vacíos que las teorías clásicas del aprendizaje han dejado, como es el caso del conductismo, el cognitivismo y el constructivismo. En la teoría del conectivismo se reconoce en este proyecto la esencia

de comprender cómo se puede dar un paso evolutivo de una sociedad de la información a una sociedad del conocimiento, a través de un aprendizaje en red.

4. Objetivos

- General

El objetivo general que se ha dispuesto para la implementación y redireccionamiento del curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal, es el siguiente: desarrollar un proceso formativo en los educandos hacia el uso y aplicación de distintos tipos de recursos didácticos y software como apoyo en los procesos de enseñanza y aprendizaje de la matemática discreta, a través de una construcción individual y social del conocimiento que promueva una conciencia crítica y constructiva, de las aplicaciones y limitaciones que tienen las tecnologías digitales y el conocimiento matemático discreto para resolver problemas en el ámbito de la informática.

- Específicos

- Utilizar distintos tipos de tecnologías digitales y software especializado como apoyo en los procesos de enseñanza y aprendizaje de la matemática discreta.
- Construir soluciones a problemas no triviales de matemática finita y sus aplicaciones en ingeniería en sistemas de información y ciencias de la computación a través del uso de distintas tecnologías digitales y software especializado.
- Identificar ventajas y desventajas del uso de tecnologías digitales y software especializado para resolver problemas contextualizados en el campo de la ingeniería en sistemas de información y las ciencias de la computación.

5. Resultados esperados

Los resultados que se espera obtener con el curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal se centran en los agentes principales involucrados con su proceso de implementación: los estudiantes, los docentes, el departamento *UNA Virtual* y la institución en general.

Los resultados esperados en un período de tiempo de tres años son los siguientes:

- Una cátedra constituida por: un coordinador general de todos los cursos tanto en sus características presenciales como virtuales, un grupo de dos profesionales con una carga de un cuarto de tiempo cada uno destinada a la investigación de las implicaciones teóricas que emergen en la práctica educativa del curso, un experto de contenido y un diseñador gráfico para la elaboración y reelaboración de materiales didácticos con una carga académica de al menos un cuarto de tiempo.
- Un equipo de cinco docentes con formación académica debidamente acreditada en el área de la tutoría virtual. Se espera que este grupo de profesores se encuentren capacitados al menos con dos cursos relacionados con la tutoría virtual y el manejo de entornos virtuales de aprendizaje.
- Un profesional del departamento *UNA Virtual* integrado debidamente con el proyecto y su coordinador, para el manejo administrativo de la plataforma y en la oferta de servicios de capacitación y consultas técnicas tanto a profesores como a estudiantes, relacionados con el uso pedagógico y técnico de la plataforma *Moodle*.
- La publicación de un artículo en una revista científica internacional cada dos años sobre los avances teóricos suscritos a las prácticas educativas en torno al curso, los alumnos, la plataforma de teleformación utilizada y las prácticas docentes presenciales y virtuales más adecuadas en el contexto de la *EI* de la Universidad Nacional de Costa Rica.
- Una mediación pedagógica de todos los materiales y recursos didácticos utilizados en la cátedra del curso tanto para el trabajo presencial como virtual.
- Capacitación continua de los docentes de la cátedra en el uso de distintos tipos de recursos didácticos digitales. Se pretende que cada docente se capacite con al menos un curso anual.
- Se promoverán acciones para crear una conciencia institucional sobre la necesidad de incorporar y diferenciar en el marco jurídico de la universidad, los derechos y obligaciones de académicos y estudiantes en el contexto de una modalidad semipresencial y/o virtual.

- Una propuesta de curso madura y validada en: prácticas de aprendizaje, estructura y estrategias de evaluación que permitan migrar la experiencia a otros cursos de la Escuela de Informática de la UNA y a futuro ofrecer servicios de formación al *Ministerio de Educación Pública* de Costa Rica en esta materia.

6. Aspectos operativos

Los aspectos operativos relacionados con este proyecto se especifican en este apartado, la exposición se divide en: administración del sistema, prácticas de aprendizaje y tecnologías, tutoría prevista, materiales didácticos y, evaluación y seguimiento.

- Administración

En el contexto de la Universidad Nacional de Costa Rica el departamento *UNA Virtual* oficialmente tiene a su cargo la administración tecnológica de la plataforma institucional, en este caso y hasta la fecha, la plataforma: *Moodle*. Bajo esta perspectiva, compete a este organismo la apertura de los cursos en toda la universidad. Sin embargo, en muchas ocasiones se ha comprobado en la práctica que esto tiene repercusiones importantes en el tiempo de creación espacial y seguimiento de cursos virtuales o semipresenciales. En este sentido, el coordinador general del curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal velará por mantener una comunicación constante con el representante del departamento *UNA Virtual* a cargo, tratando de prever inconvenientes que pongan en riesgo el inicio exitoso de la materia en sus componentes virtuales.

Los materiales del curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal serán entregados a los estudiantes a través del aula virtual institucional en el período lectivo correspondiente.

El coordinador general se encargará también de gestionar los procesos para evaluar la calidad del curso ofrecido en cuanto a: prácticas de aprendizaje, estrategias de evaluación, actualización de los contenidos, bibliografía y lincografía, calidad de la

tutoría, fortalezas y debilidades en la plataforma de teleformación, acceso a los laboratorios institucionales y rendimiento académico de los estudiantes.

- Aprendizaje y tecnologías

En el curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal el trabajo se realizará mediante la acción conjunta de profesor(a) y estudiantes, en donde se plantean y desarrollan mediante la investigación y trabajo cooperativo/colaborativo, los diferentes temas que interesa conocer. El profesor(a) cumple una función de facilitador en la construcción del conocimiento, los estudiantes son los responsables de llevar a buen término su proceso de aprendizaje a través de las actividades propuestas.

Es esencial el trabajo en equipo, la búsqueda de consenso, lluvia de ideas para obtener significados, la búsqueda de información, la elaboración de categorías y los procesos de análisis y síntesis en la adquisición y reconstrucción del conocimiento.

La propuesta del curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal se fundamenta en las siguientes actividades de aprendizaje: talleres de resolución de problemas, exposiciones grupales basadas en el empleo de tecnologías digitales y software especializado, comprobaciones de lectura, un proyecto programado, un proyecto de diseño y desarrollo de un video y, participación en foros de discusión. En la siguiente tabla se explicita en detalle el mapa de prácticas de aprendizaje correspondiente a este curso.

Actividad	Explicación	Sentido
Talleres de resolución de problemas	Es necesario que los estudiantes se integren en el trabajo de los diferentes talleres, de manera que, al finalizar cada sesión presencial o virtual, hayan propuesto soluciones a	En la materia se pretende que los estudiantes puedan usar distintos recursos didácticos para la enseñanza y el aprendizaje de la matemática discreta. Con este fin se hace

	<p>problemas no triviales relacionados con matemática discreta. Para lograrlo deberán traer revisados los materiales y recursos que el docente solicite con anticipación. Los criterios para evaluar los talleres presenciales y virtuales son los siguientes: entrega de tareas, puntualidad, completitud, observación de aula (tanto presencial como virtual), participación individual, participación en los <i>chat</i> y las <i>videoconferencias</i> de evacuación de dudas programadas por el docente, colaboración y cumplimiento de los objetivos de acuerdo con el programa del curso.</p>	<p>indispensable crear el espacio académico donde se propongan a los educandos distintos momentos de elaboración a soluciones de problemas de matemática finita a través del empleo de tecnologías digitales y software especializado, como por ejemplo el uso del programa comercial: <i>Wolfram Mathematica</i>. Durante las distintas semanas del curso se propondrá al estudiante actividades donde será necesario para su buen término, la creación crítica y reflexiva de soluciones a los problemas planteados a la luz del sustento teórico abordado de manera conjunta. Se utilizarán el <i>chat</i> y la <i>videoconferencia</i> de la plataforma <i>Moodle</i> como medios de convocatoria sincrónicos para la evacuación de dudas.</p>
--	--	--

<p>Exposiciones grupales basadas en el empleo de tecnologías y software especializado</p>	<p>Los estudiantes en subgrupos realizarán una exposición presencial sobre uno de los temas principales del curso. El tema elegido se desarrollará de manera distinta a lo contemplado por el docente en el programa de la materia, dando así con ello un aporte adicional. El objetivo principal de la exposición se fundamenta en la búsqueda de tecnologías digitales y software especializado, para la enseñanza, el aprendizaje y la investigación del área disciplinar. De forma creativa cada subgrupo pondrá en práctica el uso de los recursos y sus aplicaciones con sus demás compañeros, mediante una actividad o conjunto de actividades que así lo requieran.</p>	<p>Esta actividad pretende que los estudiantes pongan en práctica el uso de distintas tecnologías digitales y software para profundizar contenido y aplicaciones de la matemática finita. La actividad o actividades implementadas a modo de simulación por los alumnos con sus compañeros, se fundamentará en una investigación y adaptación de recursos o ideas desarrolladas por autores reconocidos a nivel nacional o internacional. Se solicitará a los alumnos participar en una <i>wiki</i> que constituirá la bitácora de producción académica desarrollada por los miembros de cada subgrupo. Allí los subgrupos adjuntarán la versión final de su trabajo y una presentación digital del mismo. Al finalizar las exposiciones se solicitará a todos visitar la <i>wiki</i> y</p>
---	---	---

		<p>analizar de manera crítica y constructiva el proceso de planificación y producción realizado por el subgrupo con la finalidad de identificar de manera colectiva las estrategias de aprendizaje y colaboración utilizadas por los miembros. También con ello se identificará socialmente aquellos estudiantes que no lograron incorporarse adecuadamente en la realización y cumplimiento de la tarea.</p>
Comprobaciones de lectura	<p>Se realizarán una serie de lecturas previas a cada sesión de trabajo (presencial y/o virtual), las cuales serán obligatorias y de ellas se aplicará una comprobación de lectura.</p>	<p>Con esta actividad hacemos explícita la importancia del estudio teórico individual y el dominio de conocimientos básicos requeridos en el campo de la matemática discreta. Todas las comprobaciones de lectura se realizarán a través de la plataforma de teleformación <i>Moodle</i>.</p>
Un proyecto programado	<p>En subgrupos de cuatro o tres estudiantes se</p>	<p>Con este trabajo se espera que los alumnos</p>

	<p>desarrollará un proyecto de programación, implementando en el lenguaje <i>Wolfram</i> alguno de los algoritmos desarrollados durante el período lectivo. El diseño del trabajo se respaldará en un proceso con las siguientes etapas: selección del algoritmo y justificación, implementación en el lenguaje de programación <i>Wolfram</i>, depuración de errores y realización de pruebas. El proyecto se presentará de manera presencial ante el grupo.</p>	<p>experimenten todas las etapas requeridas en el diseño, desarrollo e implementación de un programa en el lenguaje <i>Wolfram</i>. Además de ello, se espera como parte del proyecto un análisis crítico de las posibles aplicaciones reales del algoritmo elegido, identificando sus fortalezas y debilidades en la resolución de problemas vinculados con ingeniería en sistemas de información y ciencias de la computación. El control del avance de cada subgrupo se efectuará a través de una <i>wiki</i> donde los miembros de los subgrupos dejen huella de los procedimientos aplicados en la realización y cumplimiento de la tarea. El docente se comprometerá con cada subgrupo para realizar las observaciones pertinentes respecto a los procesos</p>
--	---	--

		<p>cognitivos observados, las estrategias de resolución de problemas aplicadas y los mecanismos meta cognitivos utilizados, permitiendo así a los estudiantes reflexionar sobre sus propios procesos de aprendizaje.</p>
<p>Un proyecto de diseño y desarrollo de un video</p>	<p>Se solicitará a los alumnos desarrollar un video de un tema libre relacionado con las áreas cognitivas del curso, donde se muestre y explique en detalle sus posibles aplicaciones e importancia científica en el campo de la ingeniería en sistemas de información y las ciencias de la computación. Se recomendará para ello el uso de software proveniente de la <i>Web 2.0</i>.</p>	<p>Con esta actividad los alumnos pondrán en práctica el diseño de materiales multimediales, analizando sus ventajas y desventajas para la comunicación científica de su especialidad. Además, los estudiantes analizarán de manera crítica la importancia del contenido matemático discreto en la resolución de problemas vinculados con su área de formación principal: la informática. Cada subgrupo llevará un registro de sus actividades en una <i>wiki</i>, mostrando sus avances y estrategias de solución (se esperaría que los alumnos a estas alturas</p>

		<p>del curso hayan mejorado las debilidades detectadas en los procesos colaborativos y meta cognitivos indicados por el docente en la realización del primer proyecto). El correo interno, el <i>chat</i> y la <i>videoconferencia</i> en la plataforma <i>Moodle</i> se utilizarán como medios para evacuar las dudas surgidas durante las semanas de realización de este trabajo.</p>
<p>Participación en foros de discusión</p>	<p>De forma individual se realizarán foros de discusión utilizando la plataforma de aprendizaje virtual <i>Moodle</i>. Los foros se evaluarán recurriendo a los siguientes criterios: cantidad y puntualidad de las participaciones (se espera como mínimo dos participaciones por alumno, la primera el día lunes o martes y la última el día jueves o viernes, el día miércoles él o la profesora moderará las</p>	<p>Los foros tendrán una duración de una semana y versarán sobre las siguientes preguntas generadoras:</p> <p>¿Por qué es importante la matemática discreta como un mecanismo instrumental en la informática?</p> <p>¿Por qué es importante la matemática discreta como un mecanismo de aplicación en la resolución</p>

	participaciones de cada estudiante y el día sábado o domingo realizará un cierre utilizando las ideas principales de la discusión), calidad de las participaciones (en términos de consistencia y profundidad a las temáticas del curso, se recomendará investigar en la Red acerca del tema) y trabajo colaborativo (las intervenciones no deberán ser aisladas, promoviendo así, la discusión respetuosa y crítica de todos).	de problemas relacionados con la informática? ¿Qué aplicaciones posee la matemática finita desde un punto de vista científico? ¿Qué aplicaciones posee la matemática discreta en la informática? ¿Cuáles tecnologías digitales y software especializado se encuentran disponibles para resolver problemas de matemática discreta?
--	---	--

- Tutoría

El grupo de profesionales que se encargarán de brindar acompañamiento a los alumnos durante las diecisiete semanas del curso estará conformado por cuatro tutores y un coordinador general. Cada tutor será responsable de un grupo con a lo sumo veinticinco estudiantes.

El tutor general (coordinador) tendrá asignadas las siguientes funciones: velar por la calidad de los servicios académicos y administrativos ofrecidos a los estudiantes, proporcionar apoyo a los docentes (sobre todo a aquellos que se inician introduciéndolos al enfoque pedagógico del curso), coordinar con la persona encargada en el departamento *UNA Virtual* la apertura y seguimiento de los cursos y procesos de capacitación y, encauzar las tareas de investigación en torno a este proyecto. Además,

el coordinador general identificará los puntos débiles en cuanto a contenido, dominio tecnológico y tutoría virtual de su grupo de colaboradores y gestionará ante las autoridades de la unidad académica el presupuesto requerido, con el objetivo de capacitar a los docentes en las áreas prioritarias.

- Materiales didácticos

Los materiales de estudio que se utilizarán en el curso *EIF-203 Estructuras Discretas para Informática* con un enfoque bimodal serán seleccionados y/o elaborados, editados y adaptados por el coordinador de la cátedra mediante la colaboración de un diseñador gráfico y los docentes del curso. Los documentos o lecturas tendrán formato *PDF*, dada su facilidad de acceso y optimización en el peso de los archivos.

En la siguiente tabla se especifica el mapa de materiales didácticos requeridos en el curso.

Material didáctico	Explicación	Sentido
Lecturas sobre los ocho tópicos principales de la materia	Se presentarán a los estudiantes los contenidos teórico-prácticos de la materia <i>EIF-203</i> mediante lecturas de naturaleza obligatoria. Los temas por tratar versarán en: recursividad, relaciones de recurrencia, análisis de algoritmos, relaciones binarias, teoría de grafos, teoría de árboles,	El material de estudio es fundamental con la intención de introducir a los estudiantes hacia la comprensión de conceptos generales, teoremas, propiedades y aplicaciones.

	máquinas y autómatas de estado finito y, lenguajes y gramáticas.	
Cuadernos interactivos para los ocho temas principales	Se proporcionará a los estudiantes ocho <i>PDF's</i> (uno por cada área temática del curso) con audio integrado explicando el contenido teórico-práctico de la materia.	Estos cuadernos interactivos poseen una navegación flexible y se constituirán en un recurso de formación complementaria durante las sesiones virtuales.
Presentaciones	Constituyen presentaciones <i>Power Point</i> o <i>Beamer</i> de todas las lecturas del curso, ofreciendo un resumen teórico-práctico de los contenidos y ejemplos a desarrollar en las clases virtuales y/o presenciales.	Serán elaboradas por los profesores de la cátedra.
Tutoriales para utilizar el software <i>Wolfram Mathematica</i> y el paquete <i>VilCretas</i>	En el curso <i>EIF-203</i> se emplea un software comercial con fines didácticos y de investigación denominado: <i>Wolfram Mathematica</i> . Estos tutoriales en formato	Ofrecerán una ayuda complementaria a la población estudiantil. Los tutoriales en formato de video se presentarán a los alumnos mediante la plataforma <i>Symbaloo</i> colgados en la red social <i>YouTube</i> .

	<p><i>PDF</i> y de video aportarán en cuanto a la utilización técnica del software y un paquete diseñado por el coordinador de la cátedra, llamado: <i>VilCretas</i>. Este paquete constituye una herramienta para la enseñanza y el aprendizaje de la matemática discreta.</p>	
<p>Glosario de comandos del software <i>Wolfram Mathematica</i> y el paquete <i>VilCretas</i></p>	<p>Los glosarios conforman un diccionario sobre los principales comandos de uso básico de <i>Mathematica</i> y todas las instrucciones que caracterizan al paquete de software <i>VilCretas</i>.</p>	<p>Cada comando incluido tendrá una descripción sobre sus funcionalidades y sintaxis empleada.</p>
<p>Archivos de apoyo del software <i>Mathematica</i> con ejemplos resueltos</p>	<p>Se ofrecerán a los estudiantes todos los ejemplos resueltos del curso <i>EIF-203</i> en sus distintas temáticas, mediante código fuente del software <i>Wolfram Mathematica</i>.</p>	<p>Este material didáctico les permitirá a los alumnos enfocarse durante las lecciones virtuales y presenciales en la profundización conceptual y aplicación de los conocimientos desarrollados, sin que medie una preocupación por reproducir lo mostrado durante la enseñanza.</p>

<p>Documentos con un formato computable (<i>CDF's</i>) para la mejoría conceptual y procedimental</p>	<p>Se ofrecerá a los estudiantes un conjunto de <i>CDF's</i>. Un <i>CDF (Computable Document Format)</i> conforma un documento de manipulación dinámica diseñado con ayuda del software <i>Wolfram Mathematica</i>. Se brindará a los alumnos distintos <i>CDF's</i> relacionados con los ocho temas principales del curso <i>EIF-203</i>.</p>	<p>Los estudiantes a través del empleo de los <i>CDF's</i> lograrán visualizar conceptos, teoremas y propiedades. Se espera que esta manipulación dinámica contribuya con una mejoría conceptual y procedimental de los contenidos.</p>
<p>Enlaces</p>	<p>Complementan software, informaciones y prácticas de los distintos temas relacionados con el curso.</p>	<p>Paquete <i>VilCretas</i>: http://www.escinf.una.ac.cr/discretas/index.php/archivos/category/7-packages Archivos de apoyo: http://www.escinf.una.ac.cr/discretas/index.php/archivos Quices interactivos: http://www.escinf.una.ac.cr/discretas/index.php/quices-interactivos Documentación del paquete <i>VilCretas</i>: https://tecdigital.tec.ac.cr/revistamatematica/Libros/</p>

		<p>Obtener una licencia del software Wolfram Mathematica: http://www.escinf.una.ac.cr/discretas/index.php/obtener-tecnologias-de-wolfram</p>
--	--	---

7. Evaluación y seguimiento del proyecto

La evaluación del proyecto *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal, integrará indicadores de calidad distribuidos en los siguientes aspectos: la gestión del curso, los materiales didácticos, los medios y tecnologías seleccionadas, la evaluación de los aprendizajes y la infraestructura que dará soporte a la implementación. Los indicadores han sido tomados como puntos de referencia del documento “Informe final” de la CONEAU (Becerra, et al., s.a.) y han sido adaptados a las características del proyecto que aquí nos ocupa, en el marco de acción de la Escuela de Informática de la Universidad Nacional de Costa Rica. Las líneas siguientes explican el contexto particular en el cuál será imprescindible efectuar los procesos de autoevaluación en las dimensiones anteriormente establecidas.

- Antes, durante y al finalizar el proyecto

El seguimiento del proyecto será ejecutado por el coordinador general del curso *EIF-203 Estructuras Discretas para Informática* en colaboración con todos los profesores de la cátedra.

En este proyecto el seguimiento se concibe como un aspecto esencial para:

- Construir, reelaborar y seleccionar nuevos materiales didácticos.
- Solucionar a breve plazo problemas de contenido, tecnológicos o de relaciones humanas.
- Garantizar una práctica educativa homogénea en cuanto al enfoque pedagógico implementado por los docentes.
- Proponer innovaciones relacionadas con las prácticas de aprendizaje y a la luz de estas nuevas propuestas definir el uso de otras herramientas tecnológicas.

- Diseñar de manera paulatina una guía didáctica que constituya una memoria escrita sobre los materiales básicos en los cuáles se fundamenta el curso, los aportes personales de cada docente (tanto en experiencias de aprendizaje como de evaluación) y ejemplos de producciones (videos o implementaciones de algoritmos) realizados por los mismos estudiantes.

Las estrategias que utilizará el coordinador se sustentarán en: un seminario permanente, un taller de reflexión pedagógica al finalizar cada período lectivo y una bitácora personal del trabajo realizado por cada uno de los docentes mediante un *blog*.

El seminario permanente será convocado por el coordinador de manera mensual y en estos espacios de trabajo se procederá a:

- Introducir a los nuevos docentes en el enfoque pedagógico del curso.
- Proponer nuevos materiales didácticos de manera conjunta.
- Proponer nuevas prácticas de aprendizaje y el uso de otras tecnologías digitales.
- Ofrecer a los docentes charlas formativas en aspectos tecnológicos y de tutoría virtual (por parte del departamento *UNA Virtual*).
- Elaborar en conjunto y durante cada semestre una guía didáctica digital con material didáctico y experiencias de aprendizaje probadas por los mismos docentes en su práctica profesional.
- Ofrecer espacios para que los estudiantes pertenecientes a la cátedra puedan compartir con otros profesores sus realizaciones creativas (esto permitirá seleccionar de forma consensuada cuáles trabajos podrían ser incluidos en la guía didáctica).

Al finalizar cada semestre el coordinador general convocará a un taller de reflexión pedagógica con la intención de:

- Identificar fortalezas y debilidades en:
 - La propuesta pedagógica del curso.
 - Las prácticas de aprendizaje y de evaluación.
 - Los materiales de estudio ofrecidos a los estudiantes.

- El equipo audiovisual y computacional disponible y sus horarios de uso.
- Identificar las fortalezas y requerimientos de la plataforma institucional *Moodle*.
- Definir las necesidades sentidas por el grupo de docentes colaboradores y a la luz de estos resultados la puesta en marcha de futuros procesos de capacitación.
- Brindar un informe sobre el trabajo realizado por la coordinación durante el período y los procesos de investigación encausados en torno al proyecto.
- Ofrecer la oportunidad para el diálogo abierto sobre el trabajo realizado por todos y todas.

Finalmente, el coordinador general abrirá un espacio privado a cada uno de los docentes del curso, quienes tendrán la responsabilidad de moderar semanalmente durante todo el semestre un *blog*, cuyos únicos invitados serán los colegas de la cátedra. El *blog* conformará una bitácora personal de los profesores sobre sus experiencias semanales con sus estudiantes ante: cada uno de los contenidos, las prácticas de aprendizaje asignadas, las experiencias de evaluación, los materiales utilizados, los materiales desarrollados por los mismos estudiantes, las dificultades presentadas en la tutoría y las necesidades sentidas junto con sus posibles soluciones. El *blog* permitirá al grupo de profesores comunicar y socializar durante el proceso, aspectos tanto positivos como negativos que serán idealmente retroalimentados por todos de manera oportuna. Es reconocible en esta iniciativa, el efecto mariposa que podría ocasionar, pues si de previo el docente considera el *blog* como un medio de control por parte del coordinador general, perderá su sentido didáctico y colaborativo.

- Indicadores de evaluación de cada aspecto operativo
 - i. Gestión del curso

El curso *EIF-203 Estructuras Discretas para Informática* como en algún momento ya fue explicado, contará con un coordinador general. Con el objetivo de garantizar el buen funcionamiento de su gestión académica y administrativa, será indispensable que la persona más próxima de acuerdo con la jerarquía organizacional, realice una evaluación sobre el cumplimiento de sus principales funciones, las cuáles se pueden resumir en:

- Cuidar la calidad de los servicios académicos y administrativos ofrecidos a los estudiantes.
- Brindar apoyo a los docentes en asuntos tanto académicos como administrativos.
- Realizar procesos de inducción para aquellos profesores que se inician con el enfoque pedagógico del curso.
- Coordinar con la persona encargada en el departamento *UNA Virtual* la apertura de cursos y procesos de capacitación.
- Analizar las fortalezas y debilidades de la plataforma institucional.
- Liderar los procesos de investigación en torno al proyecto.
- Liderar los procesos de mejoramiento de la cátedra en cuanto a contenido, prácticas de aprendizaje, experiencias de evaluación, dominio tecnológico y tutoría virtual.
- Gestionar ante las autoridades de la unidad académica el presupuesto requerido con el objetivo de capacitar a los docentes en las áreas prioritarias.

La dirección de la Escuela de Informática de la UNA será la responsable de velar por el cumplimiento de estas tareas, en la figura del coordinador general del curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal.

ii. Materiales didácticos

Los materiales didácticos seleccionados y elaborados para la cátedra del curso *EIF-203 Estructuras Discretas para Informática*, serán el resultado de un proceso continuo y colaborativo, por parte de los docentes/tutores quienes tendrán la responsabilidad de elaborar y/o adaptar estos recursos a sus distintas necesidades. En conjunto se diseñará una guía didáctica que integrará todos los materiales tanto generales como específicos, esta guía servirá como un hilo conductor de contenido y experiencias de aprendizaje para futuras promociones del curso. Con todo ello el proceso de evaluación de estos materiales a la luz de la percepción de los estudiantes deberá contemplar los siguientes aspectos:

- Los materiales son apropiados para la educación a distancia.

- Los materiales son claros y tienen una relación directa con los objetivos de aprendizaje del curso.
- Los materiales hacen referencia a aplicaciones concretas en el área temática en estudio.
- Tienen una presentación que incita a su lectura.
- Ofrecen información completa pero no acabada, mostrando al lector otras fuentes de información a materiales complementarios.

Los docentes que conforman la cátedra tendrán un lugar preponderante en los procesos de autoevaluación dirigidos a mejorar la calidad de los materiales que caracterizan al curso, por tanto, el coordinador general al finalizar cada período lectivo asumirá la responsabilidad de convocar a todos sus colaboradores para analizar los resultados obtenidos en las evaluaciones de la percepción estudiantil e iniciar las acciones que en conjunto conducirán a una reelaboración y/o re selección de los materiales.

8. Cronograma para la ejecución del proyecto

Mes N°	Actividad	Producto esperado
1-2	Definición del perfil de profesor que conformará la cátedra del curso <i>EIF-203 Estructuras Discretas para Informática</i> con un enfoque bimodal	Documento con la descripción del perfil
3	Diagnóstico en la cátedra de profesores de la materia <i>EIF-203</i> sobre requerimientos de capacitación en entornos virtuales de aprendizaje, tecnologías digitales y software especializado	Documento con recomendaciones de capacitación por parte del coordinador de la cátedra para el cuerpo docente, identificando cursos nacionales e internacionales arrojados por el diagnóstico, además, de los ofrecidos por el programa ICAI de la UNA.

		<p>Se gestionará ante la dirección de la Escuela de Informática y el departamento <i>UNA Virtual</i> el financiamiento necesario para comenzar con el proceso de capacitación correspondiente.</p>
<p>4-12</p>	<p>Capacitación docente y elaboración de material didáctico</p>	<ul style="list-style-type: none"> ▪ Los docentes de la cátedra se capacitarán en los campos de: entornos virtuales de aprendizaje, tecnologías digitales y software especializado. ▪ Paralelamente se iniciará un minucioso proceso de selección y elaboración de material didáctico en cuanto a: lecturas sobre los tópicos del curso, cuadernos interactivos, presentaciones para los contenidos de la materia, tutoriales de uso de software, glosarios de comandos, archivos de apoyo generados en <i>Wolfram Mathematica</i> y diseño de <i>CDF's</i>. Se espera en esta etapa poder contar con la participación del departamento <i>UNA Virtual</i> y un diseñador

		gráfico. El producto principal será una guía didáctica para la materia.
13-18	Planificación y diseño de las actividades de enseñanza y aprendizaje	El coordinador de la cátedra en colaboración con su grupo de profesores planificará y diseñará: talleres de resolución de problemas, exposiciones grupales de un tópico de interés del curso, comprobaciones de lectura en la plataforma <i>Moodle</i> , un proyecto de programación en el lenguaje <i>Wolfram</i> , un proyecto de elaboración de un video y al menos cuatro foros de discusión a través del aula virtual.
19-24	Conformación de un grupo de investigadores	El coordinador conformará un grupo de dos a tres investigadores para identificar ventajas, desventajas y necesidades de una modalidad <i>blended-learning</i> en el contexto de la UNA y particularmente de la Escuela de Informática. Además, este grupo se encargará de sistematizar las implicaciones teóricas y prácticas de la implementación del curso <i>EIF-</i>

		203 bajo una modalidad bimodal.
25	Pronunciamento legal	Se solicitará a la dirección de la unidad académica un pronunciamiento para regularizar en el marco jurídico de la UNA los derechos y obligaciones de académicos y estudiantes en el contexto de una modalidad semipresencial y/o virtual.
26-30	Implementación del curso <i>EIF-203 Estructuras Discretas para Informática</i> bajo una modalidad bimodal	A raíz del marco del proceso de planificación previo se iniciará una prueba piloto de un ciclo lectivo, impartiendo el curso <i>EIF-203</i> bajo una modalidad bimodal involucrando a todos los actores participantes: los estudiantes y los docentes pertenecientes a la cátedra.
31-36	Sistematización de la prueba piloto	El grupo de investigadores y profesores de la cátedra del curso <i>EIF-203</i> bajo una modalidad bimodal, sistematizarán los resultados de la prueba piloto a través de la escritura de un artículo científico. Los resultados del proceso servirán para rediseñar materiales didácticos

		<p>y actividades de aprendizaje con miras a futuras versiones del curso. También, se identificarán debilidades y fortalezas de la plataforma institucional <i>Moodle</i> con la intención de robustecer su desarrollo a las necesidades particulares de esta cátedra. Además, la experiencia servirá de base con el objetivo de iniciar un proceso de prestación de servicios dirigido a la comunidad nacional y al <i>Ministerio de Educación Pública</i> de Costa Rica en materia de tutoría virtual y complejidad de los entornos de aprendizaje virtuales. Finalmente, se iniciará con un seminario permanente, un taller de reflexión pedagógica y una bitácora personal sobre el trabajo realizado por cada docente.</p>
--	--	--

9. Presupuesto

En miles	Año 1	Año 2	Año 3	Total
Rubros				
Servicios personales (honorarios docentes)	\$ (US) 36	\$ (US) 36	\$ (US) 36	\$ (US) 108

Servicios no personales (cuerpo administrativo)	\$ (US) 15	\$ (US) 15	\$ (US) 15	\$ (US) 45
Materiales y suministros	\$ (US) 1	\$ (US) 1	\$ (US) 1	\$ (US) 3
Infraestructura y equipo	\$ (US) 50	\$ (US) 0	\$ (US) 50	\$ (US) 100
Total general	\$ (US) 102	\$ (US) 52	\$ (US) 102	\$ (US) 256

10. Bibliografía

- Becerra, M., Bavio, E. y otros (s.a.). Comisión asesora de educación a distancia. Informe final. Argentina: Ministerio de Educación, Ciencia y Tecnología.
- Escuela de Informática (2018). Misión y visión. Recuperado de: <http://www.escinf.una.ac.cr>
- Duart, J. y Sangrà, A. (2000). *Aprender en la virtualidad*. España: Gedisa.
- Espiro, S. (2008a). Aprendizaje. Especialización en Entornos Virtuales de Aprendizaje. Virtual educa.
- Espiro, S. (2008b). Aprendizaje adulto. Especialización en Entornos Virtuales de Aprendizaje. Virtual educa.
- García, A. (1998). Indicadores para la evaluación de la enseñanza en una universidad a distancia. *Revista Iberoamericana de Educación a Distancia*, 1(1), 63-85.
- Juárez, A. y Assad, C. (2001). Gestión y evaluación de un sistema a distancia. *Red de Educación a Distancia de las Universidades Nacionales Región Centro-Oeste*, 1(1), 14-18.
- Prieto, D. (2008a). Planificar para construir futuro. Especialización en Entornos Virtuales de Aprendizaje. Virtual educa.
- Prieto, D. (2008b). Fase de planificación operativa. Especialización en Entornos Virtuales de Aprendizaje. Virtual educa.
- Prieto, D. (2008c). Evaluación y seguimiento. Especialización en Entornos Virtuales de Aprendizaje. Virtual educa.

- Prieto, D. y Van de Pol, P. (2006). *E-learning comunicación y educación. El diálogo continúa en el ciberespacio*. Costa Rica: Radio Nederland Training Centre.
- Siemens, G. (2004). Conectivismo: Una teoría de aprendizaje para la era digital. Recuperado de: <http://clasicas.filos.unam.mx/files/2014/03/Conectivismo.pdf>
- Vílchez, E. y González, E. (setiembre-febrero, 2014). Percepción estudiantil sobre una metodología asistida por computadora en las áreas cognitivas del álgebra lineal y la matemática discreta. *Revista Digital Matemática, Educación e Internet*, 14(1), 1-16. doi: <http://dx.doi.org/10.18845/rdmei.v14i1.1565>
- Vílchez, E. (mayo-agosto, 2014). Estrategias de enseñanza para el curso *EIF-203 Estructuras discretas para informática* a través del uso de las redes sociales *Facebook* y *Twitter*. *Revista Electrónica Educare*, 18(2), 39-70. doi: <http://dx.doi.org/10.15359/ree.18-2.3>
- Vílchez, E. (febrero, 2017). *Matemática discreta con Mathematica a través del uso del paquete VilCretas*. Trabajo presentado en el III Simposio Internacional en Matemática Educativa (III SIME), Costa Rica, Universidad de Costa Rica. Resumen recuperado de: <http://www.cimpa.ucr.ac.cr/sime/es/>
- Universidad Nacional de Costa Rica. (2007). Modelo pedagógico. Recuperado de: <http://www.una.ac.cr/index.php/acerda-de/estrategia-institucional/2012-10-02-15-21-57>

DESARROLLO DEL PROYECTO

1. Nombre del curso virtual

EIF-203 Estructuras Discretas para Informática

2. Selección y justificación de las herramientas tecnológicas

La Universidad Nacional de Costa Rica (UNA) posee dentro de su estructura de gestión académica un departamento encargado de administrar las distintas aulas virtuales institucionales que se ofrecen a los alumnos y todos los cuerpos docentes de las unidades académicas que conforman el centro educativo. Además, este departamento denominado *UNA Virtual*, es gestor del diseño y promoción de cursos de capacitación en entornos virtuales de aprendizaje bajo la dirección de la *Vicerrectoría de Docencia* universitaria.

Oficialmente *UNA Virtual* posee dentro de sus competencias fiscalizar y brindar asesoría ante cualquier iniciativa de innovación pedagógica institucional que contemple el uso de tecnologías de información y comunicación (TIC). El departamento desde hace algunos años tomó la decisión de asumir desde un punto de vista tecnológico, los sistemas de aula virtual de la UNA basados en el uso de la plataforma de teleformación *LMS (Learning Management System): Moodle*. Por esta razón, en el marco de desarrollo del presente proyecto de intervención se empleará por la normativa institucional de la Universidad Nacional de Costa Rica esta plataforma, al darse la apertura del curso de interés en uno de los cuatro sistemas de aula virtual con los que cuenta el departamento *UNA Virtual*.

Al respecto, *UNA Virtual* administra cuatro sistemas de aula virtual (todos implementados en la plataforma *Moodle*) con distintos fines educativos: el aula virtual institucional usada por estudiantes regulares de la universidad a nivel de grado y posgrado, el aula virtual de educación permanente ofrecida para cursos de educación continua, el aula virtual de actualización profesional enfocada en procesos de capacitación, actualización y talento humano y el aula virtual de investigación y desarrollo donde se pueden realizar proyectos de investigación. En este último espacio se ha creado el curso titulado: *EIF-203 Estructuras Discretas para Informática* donde se plasmará la implementación de un entorno virtual de aprendizaje para una materia del plan de estudios de la carrera

Ingeniería en Sistemas de Información de la UNA, dando un enfoque *blended learning* que se espera a futuro logre impulsar una oferta educativa favoreciendo el desarrollo de competencias relacionadas con el trabajo colaborativo, la investigación y el auto aprendizaje.

Moodle constituye un ambiente de teleformación que facilita la gestión de procesos de aprendizaje. Es una plataforma *LMS* de distribución libre lo cual la ha convertido en una herramienta de primera elección para diversas instituciones educativas a nivel mundial. Fue creada por *Martin Dougiamas* en el año 2002 con una orientación pedagógica constructivista, bajo los principios de concebir el aprendizaje no solo como un proceso de construcción individual, sino también, como resultado de una interacción colectiva. En este sentido, incorpora distintas herramientas de comunicación sincrónica y asincrónica mediante las cuales es posible fomentar experiencias de aprendizaje innovadoras concordantes con las nuevas tendencias educativas basadas en la *Web 2.0*, o bien, en la *Web* semántica. El uso de aplicaciones móviles de realidad aumentada, la computación en la nube, la utilización de contenido abierto, el aprendizaje basado en juegos/videojuegos, los ambientes de realidad virtual y las metodologías de aprendizaje contemporáneas como la gamificación y el aprendizaje invertido (*flipped classroom*), conforman hoy por hoy realidades educativas híbridas, que pueden ser organizadas y gestionadas a través del uso de una plataforma *LMS* y particularmente el empleo de *Moodle*.

Desde una perspectiva tecnológica general en *Moodle* es posible encontrar las siguientes prestaciones:

- Inicio: permite a los estudiantes volver a la página principal del curso.
- Calificaciones: proporciona una herramienta automatizada para dar seguimiento continuo a los avances académicos de los alumnos. La retroalimentación del proceso evaluativo que normalmente posibilita es tanto cualitativa como cuantitativa.
- Mensajería o correo electrónico: les permite a los estudiantes acceder a un listado de todos los participantes y del tutor del curso, contando con un servicio de

mensajería interno para comunicarse directamente con el profesor tutor y los compañeros inscritos.

- Calendario de eventos: aparecen publicados todos los eventos vinculados al curso.
- Foros de discusión: facilita el abordaje de temas de discusión asociados a la materia mediante un mecanismo de comunicación que no ocurre en tiempo real (asíncrono).
- *Chat*: es un espacio que permite establecer de modo *online* una conversación sincrónica con otros participantes.
- Recursos y enlaces de interés: son herramientas que permiten presentar recursos o enlaces a temáticas relacionadas con la materia.
- Glosario o *wiki*: incorpora y organiza las definiciones de los principales conceptos y contenidos académicos asociados con el curso.
- Ayuda: presenta una serie de explicaciones y ejemplos para utilizar el aula virtual.

Los recursos y actividades disponibles en la actualidad dentro de la plataforma institucional *Moodle* de la Universidad Nacional de Costa Rica (en su última versión: *Moodle 3.3*), específicamente son los siguientes:

- Actividades:
 - Asistencia: permite al docente tomar asistencia en clase y a los alumnos visualizar su registro. El estado de asistencia se marca como: presente, ausente, retraso o falta justificada. Se puede personalizar a las necesidades del curso.
 - Base de datos: facilita la creación, mantenimiento y búsqueda de información en un repositorio de registros, como por ejemplo: colección de enlaces *Web*, libros, reseñas de libros, referencias de revistas, fotos o carteles.
 - *Chat*: permite abrir un espacio de diálogo en formato textual de manera sincrónica. Las sesiones de *chat* se pueden guardar y hacer públicas posteriormente a todos los participantes.

- Consulta: permite al profesor arrojar una pregunta al grupo con posibles opciones de respuestas. Este tipo de actividad se emplea para efectuar una encuesta rápida, comprobar comprensión de un tema puntual o facilitar la toma conjunta de decisiones.
- Cuestionario: es un tipo de herramienta que provee recursos para diseñar y construir cuestionarios con preguntas de: opción múltiple, verdadero/falso, coincidencia, respuesta corta y respuesta numérica. Las preguntas se pueden mostrar a los estudiantes de manera aleatoria respecto a un banco de ejercicios desarrollado previamente. Las notas son consignadas en la sección de calificaciones del aula virtual.
- Encuesta: facilita la generación de encuestas para obtener información de los alumnos.
- Encuestas predefinidas: proporciona una serie de instrumentos predeterminados que el docente puede emplear para conocer la efectividad de su enseñanza a juicio de los estudiantes o profundizar las características de la población participante del curso.
- Foro: abre espacios de discusión asincrónicos en períodos de tiempo definidos por el profesor. Hay varios tipos de foros según las necesidades de la actividad a planificar, por ejemplo, discusiones con: temas abiertos o tema único.
- Glosario: los participantes pueden crear y administrar una lista de definiciones en analogía al funcionamiento de un diccionario. Los alumnos tendrán la posibilidad de adjuntar archivos y la información se mostrará en orden alfabético. La actividad se puede programar como parte de la sección de calificaciones del aula.
- Herramienta externa: permite a los estudiantes utilizar recursos y actividades afiliadas en páginas *Web* externas. No son enlaces a sitios de *Internet* pues soportan la inclusión de calificaciones en la sección del aula virtual correspondiente. Normalmente se necesita un proveedor de herramienta *LTI (Learning Tools Interoperability)*.

- Lección: facilita al docente presentar contenidos o actividades del curso de una forma distinta y flexible, creando un conjunto de páginas de contenido con actividades educativas, como por ejemplo, preguntas de: opción múltiple, respuesta corta y correspondencia. El trabajo realizado por los alumnos podrá ser calificado formalmente a través del libro de calificaciones del aula.
- Paquete *SCORM*: son paquetes que proveen objetos de aprendizaje que es posible desplegar en *Moodle* para mostrar contenidos multimedia y/o animaciones, o bien, realizar pruebas escritas con muy diversos formatos.
- Taller: permite la revisión por pares (anónima o no) del trabajo realizado por los estudiantes, donde se podrá adjuntar cualquier tipo de contenido digital: documentos, hojas de cálculo, imágenes, entre otros. Es una interesante herramienta para ejercitar en la población participante procesos de coevaluación. Todo queda registrado en la sección de calificaciones del aula virtual.
- Tarea: facilita la creación de tareas a evaluar por parte del docente, en las cuales los alumnos tendrán la posibilidad de adjuntar cualquier tipo de archivo. Se les brinda a los estudiantes un campo de texto con la intención de complementar la información de sus envíos digitales y el profesor posteriormente evaluará el trabajo realizado dando una retroalimentación con escalas personalizadas, archivos con correcciones, o bien, el uso de una rúbrica. La nota queda consignada en la sección de calificaciones.
- *Wiki*: se utiliza para editar una colección de páginas *Web* de manera individual o colaborativa dejando un histórico de las versiones previas. Es una herramienta ideal de seguimiento en el contexto de una actividad educativa.
- Recursos:
 - Archivo: le permite al docente compartir con los alumnos información digital en el formato que se desee.
 - Carpeta: muestra un grupo de archivos organizados dentro de una carpeta. Los archivos se adjuntan en el interior de este recurso.

- Etiqueta: facilita la incorporación de texto (títulos, subtítulos, descripciones) y objetos multimedia (videos o audio) entre otros recursos y actividades de la materia. Contribuyen con un mejoramiento de la apariencia estética del aula virtual.
- Libro: brinda la opción de crear materiales con un formato de libro, subdivido en capítulos y subcapítulos con la posibilidad de incluir información en diversidad de formatos: texto, imagen, video y audio.
- Paquete de contenido *IMS*: añade dentro del aula virtual paquetes de contenido *IMS Content Packaging*.
- Página: permite al profesor la creación de páginas *Web* mostrando texto, imágenes, video, sonido, enlaces y código incrustado (por ejemplo, para la inserción de un mapa de *Google* o una presentación *Prezi*).
- URL: crea enlaces a sitios externos de *Internet*.

El entorno virtual de aprendizaje dirigido a la materia *EIF-203 Estructuras Discretas para Informática* se sustentará en un aula virtual *Moodle* con una distribución por temas como se detalla a continuación: generalidades, recursividad, relaciones de recurrencia, análisis de algoritmos, relaciones binarias, teoría de grafos, teoría de árboles, máquinas y autómatas de estado finito y, lenguajes y gramáticas. Cada uno de los temas indicados hace referencia a una unidad de contenido principal del curso *EIF-203*. Se ha tomado la decisión de presentar la materia bimodal con un formato temático y no semanal, con el objetivo de agrupar las distintas actividades por área de contenido, organizando visualmente la información al estudiante de manera cronológica con los distintos recursos seleccionados y creados. En la sección generalidades se brindarán a los alumnos descripciones globales de la materia, un cronograma de trabajo detallado, recursos y enlaces que servirán de soporte durante el transcurso de todo el ciclo lectivo. En las otras secciones se irán compartiendo las distintas actividades y recursos didácticos específicos para cada área de contenido.

En función del mapa de aprendizaje planificado para el curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal, las herramientas a utilizar del aula virtual institucional en la plataforma *Moodle* son las siguientes:

- Creación de tareas para la generación de una serie de talleres de resolución de problemas ante situaciones no triviales vinculadas con la matemática discreta. Paralelamente a la construcción de este espacio de entrega de soluciones inéditas, se utilizará una *wiki* con el objetivo de evidenciar el trabajo colaborativo efectuado en las propuestas de resolución y, creación crítica y reflexiva, llevando un histórico del nivel de logro alcanzado y la evolución de los aprendizajes mostrados por los participantes.
- Creación de una tarea para la entrega de un documento y una presentación exponiendo un tema principal del curso de una forma novedosa, mostrando el uso de distintas tecnologías digitales y software con la intención de profundizar aplicaciones de la matemática finita. Se habilitará la herramienta de glosario a cada subgrupo con la intención de construir de manera colaborativa un conjunto de definiciones base de consulta. Además, el proceso de desarrollo del documento escrito y la presentación se evidenciará a través del empleo de una *wiki*, consignando una bitácora de producción académica de todos los miembros del equipo. Al finalizar cada exposición se solicitará a los demás subgrupos valorar el proceso de planificación y producción realizado por los compañeros expositores, incentivando una coevaluación al activar la herramienta de taller en *Moodle*.
- El docente compartirá distintas lecturas de naturaleza teórico-práctica, exhibiendo el contenido esencial de la materia en cuanto a: definiciones, propiedades, teoremas, ejemplos y aplicaciones. Los archivos serán subidos en formato *PDF* por su facilidad de uso y optimización con respecto a su peso informático. Con la intención de garantizar la revisión previa de estos documentos se planificará por lectura, un cuestionario en el aula virtual que permitirá evaluar la temática tratada (algunas evaluaciones serán paquetes *SCORM* creados con el software *Articulate Studio*). Se reconoce con ello, la importancia del aprendizaje receptivo en un curso de matemática discreta y el condicionamiento operante propuesto por Skinner.

- Se facilitará a los alumnos enlaces con ejemplos de algoritmos relacionados con la materia, en lenguajes de programación distintos al lenguaje *Wolfram*. Por subgrupos, se solicitará desarrollar un proyecto de programación implementando en *Wolfram* uno de los algoritmos ejemplificados. La tarea será creada en el espacio del aula virtual institucional para su entrega. Además, se abrirá una *wiki* de análisis crítico de las posibles aplicaciones reales del algoritmo elegido, identificando fortalezas y debilidades como recursos de mediación vinculados con problemas de ingeniería en sistemas de información y ciencias de la computación.
- Se creará una tarea para la entrega de una producción audiovisual en formato de video que los estudiantes subirán a una red social (por ejemplo: *YouTube* o *Vimeo*). El video mostrará de manera creativa e innovadora un tema libre vinculado con el curso *EIF-203*, exponiendo aplicaciones e importancia científica en el campo de la ingeniería en sistemas de información y las ciencias de la computación. Los registros de las actividades (avances y estrategias de resolución) llevadas a cabo por cada subgrupo, quedarán plasmados en una *wiki* instaurada para estos fines.
- Se concebirán al menos tres foros de discusión con las siguientes preguntas generadoras:
 - ¿Por qué es importante la matemática discreta como un mecanismo instrumental en la informática?
 - ¿Qué aplicaciones posee la matemática discreta en la informática?
 - ¿Cuáles tecnologías digitales y software especializado se encuentran disponibles para resolver problemas de matemática discreta?
- Todos los archivos compartidos serán distribuidos en carpetas con identificadores adecuados, logrando con ello, un ambiente claro en cuanto a la ubicación de la información ofrecida por el profesor en el aula virtual.
- Las etiquetas servirán de base para mejorar la apariencia del aula virtual en cada una de sus secciones (temas), incorporando: texto, imágenes y/o videos.
- Se emplearán el *chat* y la *videoconferencia* de la plataforma para propiciar distintos espacios de encuentro no presencial como medios de convocatoria sincrónicos hacia la evacuación de dudas de manera horizontal: profesor-

estudiantes, estudiantes-estudiantes. Además, se incentivará el uso de la mensajería interna como un recurso adicional para el abordaje de las consultas surgidas en cada una de las actividades a realizar durante el período lectivo.

3. Planificación de las clases

Clase 1

1. Título

Los grafos... más allá de un conjunto de puntos y aristas

2. Objetivos

- Comprender los conceptos principales asociados a la teoría de grafos: definición de grafo, adyacencia, incidencia, tipos de grafos, concepto de valencia, ruta, circuito, circuito y camino de *Euler*, circuito y camino de *Hamilton* y, grafo conexo.
- Utilizar software como recurso de apoyo para profundizar los conceptos principales vinculados con la teoría de grafos.

3. Contenidos de la clase

- Concepto de grafo, digrafos y sus propiedades.
- Tipos de grafos.
- Caminos y circuitos: aplicaciones en informática.

4. Bibliografía

- Ávila, J. (2005). *Estructuras de matemática discreta para computación*. Manuscrito no publicado, Escuela de Informática – Universidad Nacional de Costa Rica. Heredia, Costa Rica. Recuperado de <https://drive.google.com/file/d/1D4EfxRGTOc3VbAuPNhIaXpi8gZ6hXlap/view>
- Vílchez, E. (2017). *VilCretas Grafos*. España: Symbaloo. Recuperado de <https://www.symbaloo.com/mix/vilcretasgrafos>
- Vílchez, E. (2018). *Unidad Didáctica: Teoría de Grafos*. Unidad 1, Fase 4. Manuscrito no publicado, Aprende Virtual – Universidad Técnica Nacional. Alajuela, Costa Rica.

- Trigueros, E. (2015). *Material segunda parte: Teoría y práctica*. Manuscrito no publicado, Escuela de Informática – Universidad Nacional de Costa Rica. Heredia, Costa Rica. Recuperado de <https://drive.google.com/file/d/1SbThBx5rrN7qNVBeLVjmlskOYqK80BVR/view>

5. Recursos multimedia

- Imagen: Profesor. Descripción: imagen del docente del curso. URL: <https://photos.app.goo.gl/Ziq0swG8uCOYzf8f2>. Este recurso se empleará también en las otras planificaciones.
- Imagen: Grafo_ejemplo_5_conecsi. Descripción: imagen de un grafo sobre un mapa simulando recorridos entre varias ciudades. URL: https://es.m.wikipedia.org/wiki/Archivo:Grafo_ejemplo_5_conecsi.png
- Video: ¿Qué tienen que ver Andrés Iniesta, Tyrion Lannister y tus amigos de Facebook? | Teoría de grafos. Descripción: video introductorio del matemático *Eduardo Sáenz de Cabezón* sobre la teoría de grafos. URL: <https://youtu.be/lp-1rvtRYQg>
- Imagen: Complete_graph_K7.svg. Descripción: se muestra una figura que puede ser dibujada sin levantar el lápiz del papel. URL: https://en.wikipedia.org/wiki/File:Complete_graph_K7.svg
- Imagen: 440px-Petersen1_tiny.svg. Descripción: comparte una figura que no puede ser dibujada sin levantar el lápiz del papel. URL: https://en.wikipedia.org/wiki/File:Petersen1_tiny.svg
- Imagen: portada.jpg_1951930094. Descripción: imagen de la galería *Imaginary* perteneciente al *Instituto Matemático de Oberwolfach*, ubicado en Alemania. URL: <http://www.parqueciencias.com/parqueciencias/historico/exposiciones/imaginary.html>
- Imagen: Firma. Descripción: firma del profesor de la materia. URL: <https://photos.app.goo.gl/7GFDDNokZNVYPEGu2>. Este recurso se compartirá con las otras planificaciones.

- Imagen: NetworkTopology-Star. Descripción: imagen de un grafo estrella. URL: <https://commons.wikimedia.org/wiki/File:NetworkTopology-Star.png>. Este recurso se compartirá con las otras planificaciones.

6. Actividades

Las actividades propuestas se basan en un trabajo colaborativo a través de una *wiki* dentro de la plataforma *Moodle* y una defensa sincrónica en un punto de reunión en el metaverso *Second Life*. A continuación, se detallan sus características:

- Consigna de la *wiki*: en este espacio de trabajo colaborativo investigarán teóricamente y de forma práctica los conceptos de: grafo, adyacencia, incidencia, tipos de grafos, valencias, ruta, circuito, camino de *Euler*, circuito y camino de *Hamilton* y, grafo conexo.
- Objetivos:
 - Comprender los conceptos básicos relacionados con la teoría de grafos.
 - Emplear software especializado para profundizar la terminología elemental propia de la teoría de grafos.
 - Utilizar un mundo de realidad virtual como escenario de debate de ideas y exploración de nociones matemáticas.
 - Desarrollar habilidades de aprender a aprender, saber hacer y aprender en compañía de otros.
- Plazo: 5 días.
- Evaluación: contempla los criterios especificados en la rúbrica correspondiente a la evaluación de los “*talleres de resolución de problemas*”, consignada en la *Guía Didáctica* del curso. A saber:
 - Creación crítica y reflexiva de las soluciones: los estudiantes resuelven de manera creativa e inédita el documento colaborativo.
 - Uso de recursos y software especializado: los alumnos utilizan recursos didácticos y software como mecanismos de construcción y consulta.

- Completitud del contenido: el documento colaborativo es completo y da una respuesta creativa y puntual.
- Organización y secuencia: los estudiantes presentan de forma organizada el contenido, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
- Originalidad: el texto colaborativo es original, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.
- Uso del tiempo: el documento colaborativo se realiza en el tiempo dispuesto por el docente y la defensa en el metaverso *Second Life* respeta los tiempos asignados por el moderador.
- Calidad de las participaciones en la *wiki*: las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.
- Participación en el foro: los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio de los temas y una intervención constructiva.
- Planificación previa y trabajo en clase: los estudiantes planifican previamente junto al profesor el diseño y desarrollo del documento colaborativo. El trabajo en clase resuelto por los alumnos tiene un buen desempeño.
- Dominio en el uso del software *Wolfram Mathematica*: los alumnos demuestran dominio en el empleo del software *Wolfram Mathematica*, específicamente en la construcción de grafos y resolución de circuitos, circuitos y rutas de *Euler*, circuitos y rutas de *Hamilton* y cálculo de valencias.

7. Foro

- Objetivos:

- Analizar las aplicaciones de la teoría de grafos en la ingeniería en sistemas de información o ciencias computacionales.

- Analizar las aplicaciones de la teoría de grafos en otras áreas de naturaleza científica.
- Comprender la utilidad de la teoría de grafos.
- Consigna: este espacio tiene como propósito analizar las aplicaciones de la teoría de grafos en la ingeniería en sistemas de información o ciencias computacionales, además, de otras áreas de carácter científico. Investigue dichas aplicaciones y comparta sus conclusiones mostrando al menos un ejemplo particular.
- Plazo de participación: 7 días.

Clase 2

1. Título

La molécula de *BUCKYBALL* y los grafos

2. Objetivos

- Comprender y aplicar los teoremas y algoritmos para determinar la existencia y trayectorias de circuitos de *Euler* y de *Hamilton* sobre un grafo.
- Analizar cómo se construyen las matrices de adyacencia e incidencia en un grafo y sus propiedades.

3. Contenidos de la clase

- Circuitos y rutas de *Euler*, algoritmos, teoremas y aplicaciones.
- Circuitos y rutas de *Hamilton*, teoremas relacionados (*Dirac* y *Ore*) y aplicaciones.
- Matriz de adyacencia e incidencia y sus propiedades.

4. Bibliografía

- Ávila, J. (2005). *Estructuras de matemática discreta para computación*. Manuscrito no publicado, Escuela de Informática – Universidad Nacional de Costa Rica. Heredia, Costa Rica. Recuperado de <https://drive.google.com/file/d/1D4EfxRGTOc3VbAuPNhlaXpi8gZ6hXlap/view>

- Herráez, A. (2011). Las asombrosas estructuras del carbono: fullerenos, grafenos y nanotubos. *Revista Apuntes de Ciencias*, 1(1), 22-25. Recuperado de http://biomodel.uah.es/personal/2011/asombrosas_estructuras_carbono_con3D.pdf
- Vasilievna, K. y Ortiz, U. (2002). La estructura del fullereno C60 y sus aplicaciones. *Revista Ciencia UANL*, 5(4), 475-479. Recuperado de <http://www.redalyc.org/pdf/402/40250407.pdf>
- Vílchez, E. (2017). *VilCretas Grafos*. España: Symbaloo. Recuperado de <https://www.symbaloo.com/mix/vilcretasgrafos>
- Vílchez, E. (2018). *Unidad Didáctica: Teoría de Grafos*. Unidad 1, Fase 4. Manuscrito no publicado, Aprende Virtual – Universidad Técnica Nacional. Alajuela, Costa Rica.
- Trigueros, E. (2015). *Material segunda parte: Teoría y práctica*. Manuscrito no publicado, Escuela de Informática – Universidad Nacional de Costa Rica. Heredia, Costa Rica. Recuperado de <https://drive.google.com/file/d/1SbThBx5rrN7qNVBeLVjmlskOYqK80BVR/view>

5. Recursos multimedia

- Imagen: Robert_Curl_crop_2009_CHAO. Descripción: fotografía del científico norteamericano *Robert Curl*, quien con sus investigaciones demostró la existencia de la molécula *BUCKYBALL*. URL: https://commons.wikimedia.org/wiki/File:Robert_Curl_crop_2009_CHAO.jpg
- Imagen: Harold_Kroto_1c389_8471.sweden. Descripción: fotografía de *Harold Kroto* uno de los investigadores que descubrió la molécula de *BUCKYBALL*. URL: https://commons.wikimedia.org/wiki/File:Harold_Kroto_1c389_8471.sweden.jpg
- Imagen: Richard_Smalley. Descripción: fotografía del científico estadounidense *Richard Smalley* quien ganador el premio nobel en química en el año de 1996. URL: https://upload.wikimedia.org/wikipedia/commons/a/a8/Richard_Smalley.jpg
- Video: Nanotubos de carbono y fulerenos. Descripción: video con *Richard Smalley* explicando las implicaciones de su descubrimiento relacionado con la molécula de

BUCKYBALL y sus aplicaciones en el ámbito industrial. URL: <https://youtu.be/mOzijWeC8tY>.

- Imagen: Truncatedicosahedron. Descripción: imagen que muestra la forma de un icosaedro truncado. URL: <https://commons.wikimedia.org/wiki/File:Truncatedicosahedron.jpg>
- Imagen: 124782975_a69f1de9d8_b. Descripción: imagen que muestra la molécula *C60*. URL: <https://www.flickr.com/photos/ghutchis/124782975>
- Imagen: Grafo. Descripción: código QR que proporciona una explicación en video sobre el uso del comando Grafo perteneciente al paquete de software denominado VilCretas. URL: <https://photos.app.goo.gl/Att9oneyJjrJMXOh2>

6. Actividades

Las actividades de la semana se circunscribirán en la elaboración colaborativa de un archivo en código fuente del software *Wolfram Mathematica* mediante el uso de una *wiki*, se presente a continuación la consigna de este espacio y otros detalles:

- Consigna de la *wiki*: investiguen de manera individual qué es la molécula *BUCKYBALL* y cuáles son sus principales características matemáticas. Con esta intención empleen la aplicación de realidad aumentada *Augment* (<http://www.augment.com/es>), específicamente en: *Education/Bucky-Ball*, donde es posible visualizar la molécula. Tomando como base este proceso exploratorio, reviertan en este espacio la respuesta a las siguientes preguntas: ¿qué es la molécula *BUCKYBALL*?, ¿las caras de este poliedro qué figuras forman?, ¿cuántas aristas contiene?, ¿la molécula podría considerarse un grafo?, si es así, ¿cuáles son sus características: número de nodos y lados, cálculo de valencias, entre otras?, ¿cómo se lograría encontrar el número de aristas mediante el uso de una fórmula? Utilizando los datos compartidos y recopilados construyan a través del uso del software *Wolfram Mathematica* la molécula *BUCKYBALL*. Empleando los teoremas y algoritmos de la clase y, el programa *Wolfram Mathematica* indaguen sobre la molécula *BUCKYBALL*, la existencia de una ruta y un circuito de *Euler* y de un camino y un circuito de *Hamilton*, dando respuesta

desde un punto de vista tanto teórico como práctico. Si es posible alguna de estas trayectorias se deberá dar un ejemplo. Asimismo, representarán con al menos una matriz de adyacencia y una matriz de incidencia la molécula *BUCKYBALL*. ¿Cuántas matrices de adyacencia diferentes representan esta molécula?, ¿cuántas matrices de incidencia distintas?

- **Objetivos:**
 - Comprender qué es la molécula de *BUCKYBALL* y su importancia.
 - Utilizar una aplicación de realidad virtual para visualizar las propiedades de la molécula de *BUCKYBALL*.
 - Emplear software para modelar la molécula de *BUCKYBALL*.
 - Aplicar los teoremas y algoritmos de existencia y trayectorias de circuitos *eulerianos* y *hamiltonianos* sobre la molécula de *BUCKYBALL*.
 - Construir matrices de adyacencia e incidencia sobre la molécula de *BUCKYBALL*.
 - Desarrollar habilidades de aprender a aprender, saber hacer y aprender en compañía de otros.

- **Plazo:** 7 días.

- **Evaluación:** se fundamenta en la rúbrica de valoración de los “*talleres de resolución de problemas*”, ya establecida dentro de la *Guía Didáctica* de este curso. Particularmente estos criterios son:
 - Creación crítica y reflexiva de las soluciones: los estudiantes resuelven de manera creativa e inédita la solución a las distintas preguntas arrojadas con respecto a la molécula *BUCKYBALL* y su relación con la teoría de grafos.
 - Uso de recursos y software especializado: los alumnos utilizan recursos didácticos y software como mecanismos de construcción, consulta de hipótesis y resolución.

- Completitud del contenido: el archivo colaborativo del software *Wolfram Mathematica* es completo y da una respuesta creativa y puntual a las diferentes situaciones planteadas.
- Organización y secuencia: los estudiantes presentan de forma organizada el contenido, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
- Originalidad: las soluciones construidas con apoyo de software son originales, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.
- Uso del tiempo: la conformación de los subgrupos y el archivo *.nb* del programa *Wolfram Mathematica* se realizan en el tiempo dispuesto por el docente.
- Calidad de las participaciones en la *wiki*: las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.
- Participación en el foro: los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio y una intervención constructiva.
- Planificación previa y trabajo en clase: los estudiantes planifican previamente junto al profesor el diseño y desarrollo de las soluciones integradas en el archivo *.nb*.
- Dominio en el uso del software *Wolfram Mathematica*: los alumnos demuestran dominio en el empleo del programa *Wolfram Mathematica* y del paquete de software *ViiCretas*.

7. Foro

▪ Objetivos:

- Analizar las características de la molécula *BUCKYBALL*.
- Analizar las aplicaciones de la molécula *BUCKYBALL* y los *fullerenos*.
- Desarrollar competencias digitales relacionadas con la búsqueda y selección de información y recursos en la *Web*.

- Consigna: este foro pretende profundizar el análisis de la molécula *BUCKYBALL*. Cada participante para ello seleccionará un recurso de la *Web* (imagen, nube de palabras, video, audio, presentación) que brinde un aporte adicional sobre la comprensión y aplicaciones de la molécula *BUCKYBALL* y los *fullerenos* en general. Se deberá justificar en este espacio la razón de la selección del recurso, especificando los motivos teóricos, prácticos y aplicados que hacen de él un medio de profundización de estos contenidos. Será valioso también, buscar recursos que vinculen el tema de los *fullerenos* con la teoría de grafos. El recurso se compartirá a través de la dirección *URL* respectiva o insertando el objeto directamente en la entrada del foro.
- Plazo de participación: 7 días.

Clase 3

1. Título

Los sociogramas una aplicación contemporánea de la teoría de grafos

2. Objetivos

- Comprender el concepto de sociograma.
- Comprender el concepto de grafo ponderado.
- Describir el problema del agente viajero.
- Aplicar el algoritmo de *Dijkstra*.
- Encontrar la longitud de un camino más corto sobre un problema con datos reales.

3. Contenidos de la clase

- Concepto de grafo ponderado.
- Problema del agente viajero.
- Algoritmo del camino más corto (*Dijkstra*).
- Problemas y aplicaciones clásicas de grafos (sociogramas).

4. Bibliografía

- Aguirre, J. (2011). *Introducción al Análisis de Redes Sociales*. Argentina: Documentos de Trabajo, 82, Centro Interdisciplinario para el Estudio de Políticas Públicas. Recuperado de <http://www.pensamientocomplejo.com.ar/docs/files/J.%20Aguirre.%20Introducci%C3%B3n%20al%20An%C3%A1lisis%20de%20Redes%20Sociales.pdf>
- Ávila, J. (2005). *Estructuras de matemática discreta para computación*. Manuscrito no publicado, Escuela de Informática – Universidad Nacional de Costa Rica. Heredia, Costa Rica. Recuperado de <https://drive.google.com/file/d/1D4EfxRGTOc3VbAuPNhlaXpi8gZ6hXlap/view>
- Cornejo, J. (2006). El análisis de las interacciones grupales: las aplicaciones SOCIOS. *Revista Anuario de Psicología*, 37(3), 277-297. Recuperado de <http://www.raco.cat/index.php/anuariopsicologia/article/viewFile/61842/82854>
- Romero, R. (2010). El sociograma. *Revista Innovación y Experiencias Educativas*, 1(35), 1-15. Recuperado de https://archivos.csif.es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero_35/ROCIO_ROMERO_CALVO_02.pdf
- Vílchez, E. (2017). *VilCretas Grafos*. España: Symbaloo. Recuperado de <https://www.symbaloo.com/mix/vilcretasgrafos>
- Vílchez, E. (2018). *Unidad Didáctica: Teoría de Grafos*. Unidad 1, Fase 4. Manuscrito no publicado, Aprende Virtual – Universidad Técnica Nacional. Alajuela, Costa Rica.
- Trigueros, E. (2015). *Material segunda parte: Teoría y práctica*. Manuscrito no publicado, Escuela de Informática – Universidad Nacional de Costa Rica. Heredia, Costa Rica. Recuperado de <https://drive.google.com/file/d/1SbThBx5rrN7qNVBeLVjmlskOYqK80BVR/view>

5. Recursos multimedia

- Imagen: Sociogramas. Descripción: nube de palabras de elaboración propia con los términos más importantes vinculados con la teoría de sociogramas. URL: <https://photos.app.goo.gl/oEU8jSRrKsusk4Pz1>

- Video: VIDEO SOCIOGRAMA. Descripción: presenta la definición de sociograma, su importancia, posibles usos y formas de aplicación de esta técnica. URL: <https://youtu.be/SoAh1BB9918>
- Video: Understanding sociograms. Descripción: se presenta una explicación detallada sobre la forma de construcción de un sociograma interpretado como un grafo, en una herramienta de análisis denominada: *NVivo*. Introduce, además, el término de sociograma egocéntrico. URL: <https://youtu.be/wwwyXC6IDK98>
- Video: Social Network Analysis. Descripción: ponencia presentada en el año 2014 dentro del seno de un evento académico denominado: *Wolfram Technology Conference*. La exposición explica en detalle prestaciones específicas que ofrece el software *Wolfram Mathematica* con la intención de realizar análisis de redes sociales. URL: <https://youtu.be/7zZjLBtHf7c>
- Video: AIDijkstra. Descripción: explica el funcionamiento de un comando llamado: AIDijkstra, perteneciente al paquete de software *VilCretas*, que resuelve paso a paso la aplicación del algoritmo de *Dijkstra*. URL: <https://youtu.be/8OKGR7nK3V8>
- Video: AnimarGrafo. Descripción: explica el funcionamiento de la instrucción *AnimarGrafo* del paquete *VilCretas*, utilizada para animar sobre un grafo una trayectoria. URL: <https://youtu.be/T4tl4bwCJbl>

6. Actividades

Los aspectos que describen las actividades de la semana se especifican a continuación:

- Consigna de una *wiki* a emplear: analicen cuáles son las conexiones digitales existentes entre los miembros del subgrupo, sea en *Facebook*, *Twitter* u otra red social de sugestión común, respondiendo a la pregunta: ¿qué conexiones poseen entre sí que consideren relevantes en su vida social en la *Internet*? Con la información recopilada y discutida construyan un grafo social donde cada uno constituya un nodo. Se podrán añadir nodos que representen otras personas que no estén en el subgrupo, páginas *Web*, intereses, juegos o cualquier otro aspecto de relevancia. El grafo deberá reflejar todas las conexiones posibles entre los participantes por lo que se espera posea una cantidad de nodos y aristas

relativamente significativa. Utilizando el software *Wolfram Mathematica* y la instrucción Grafo del paquete VilCretas, construyan un grafo social 3D con la información recolectada. Clasifiquen el grafo construido en alguno de los tipos vistos en clase: completo, bipartito, mariposa, rueda, estrella, círculo, camino, regular, red, entre otros. Usando los grados o valencias determinen la cantidad de aristas que lo caracterizan. Con el grafo diseñado, respondan las siguientes preguntas justificando teóricamente y a través del uso del paquete VilCretas como recurso de apoyo: ¿posee circuitos de *Euler*? y ¿posee circuitos de *Hamilton*? Ahora se reconstruirá el grafo creado asignando pesos a cada una de las aristas en función del nivel de importancia en las relaciones sociales entre los nodos. Dos nodos débiles pero vinculados con una arista, podrían tener un peso 1, por el contrario, dos nodos con una relación fuerte podrían tener asociado un valor de 100, las escalas serán definidas por cada subgrupo, explicando en detalle el criterio utilizado para ello. Posteriormente, seleccionarán dos nodos no adyacentes en su grafo social, con el objetivo de aplicar sobre ellos el algoritmo de la longitud del camino más corto, para determinar una ruta de longitud más larga usando el paquete VilCretas y el software *Wolfram Mathematica* ¿Qué representa el peso máximo de la ruta? Finalmente, se responderá a la pregunta: ¿es posible resolver el problema del agente viajero (*PAV*) en el grafo creado?, si es así, deberán devolver dicha solución y en caso contrario, modificarán el grafo con la intención de poder dar respuesta al *PAV*, explicando el criterio de alteración en el grafo original. En cualquier caso, animarán visualmente la solución utilizando la instrucción AnimarGrafo. Cada subgrupo se abocará en la elaboración de un video que explique la presentación de su grafo social tridimensional y el proceso realizado en cada una de las etapas ya descritas. El video será publicado en: *YouTube* o *Vimeo*, con una duración máxima de cinco minutos y mínima de cuatro.

- Objetivos:
 - Conocer distintos programas para el análisis de redes sociales.
 - Aplicar los conceptos, propiedades y teoremas de la teoría de grafos en sociogramas construidos con datos reales.

- Resolver el problema del agente viajero sobre un sociograma.
 - Aplicar el algoritmo de *Dijkstra* sobre un sociograma.
 - Emplear el software *Wolfram Mathematica* en la resolución del problema del agente viajero y el algoritmo de *Dijkstra*.
 - Desarrollar habilidades de diseño y desarrollo de una producción audiovisual.
 - Desarrollar habilidades de aprender a aprender, saber hacer y aprender en compañía de otros.
- Plazo: 7 días.
- Evaluación: los criterios que se detallan a continuación se sustentan en la rúbrica para evaluar los “*talleres de resolución de problemas*”, incluida dentro de la *Guía Didáctica* de la presente materia:
 - Creación crítica y reflexiva de las soluciones: los estudiantes resuelven de manera creativa e inédita las soluciones a los problemas propuestos y, el diseño y desarrollo del video.
 - Uso de recursos y software especializado: los alumnos utilizan recursos didácticos y software como mecanismos de construcción y consulta.
 - Completitud del contenido: el video elaborado y las formas de resolución en la *wiki* son completas y dan una respuesta creativa y puntual.
 - Organización y secuencia: los estudiantes presentan de forma organizada el contenido en el video y la *wiki*, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
 - Originalidad: el video es original, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.
 - Uso del tiempo: el avance del trabajo colaborativo en la *wiki* y el video se realizan en el tiempo dispuesto por el docente.
 - Calidad de las participaciones en la *wiki*: las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.

- Participación en el foro: los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio de los temas y una intervención constructiva.
- Planificación previa y trabajo en clase: los estudiantes planifican previamente junto al profesor las soluciones a los problemas propuestos y, el diseño y desarrollo del video. El trabajo en clase resuelto por los alumnos tiene un buen desempeño.
- Dominio en el uso del software *Wolfram Mathematica*: los alumnos demuestran dominio en el empleo del software *Wolfram Mathematica* y el paquete *VilCretas*.

7. Foro

- Objetivos:
 - Investigar distintas herramientas de análisis de sociogramas y redes sociales.
 - Utilizar con datos reales herramientas de análisis de sociogramas y redes sociales.
 - Identificar las características que debe poseer una herramienta eficaz para el análisis de sociogramas y redes sociales.
 - Emplear recursos de la *Web 2.0* para el análisis de sociogramas y redes sociales.

- Consigna: este foro tiene como objetivo estudiar herramientas de análisis de sociogramas. Se iniciará indagando en la *Web* distintas herramientas disponibles para el análisis de sociogramas y redes sociales. Como sugerencia se les comparte este interesante blog: <http://ars-uns.blogspot.com/2016/06/30-paquetes-de-software-de-ars.html>. Seleccionen al menos tres herramientas y publiquen en este espacio una justificación teóricamente fundamentada de sus elecciones y algunos *screen shots* como evidencia del trabajo exploratorio realizado al utilizar cada software.

- Plazo de participación: 7 días.

4. Redacción de las clases

Clase 1: Los grafos ... más allá de un conjunto de puntos y aristas

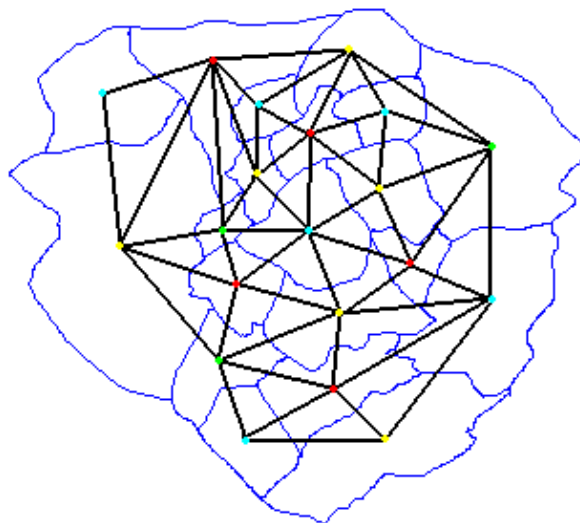


Enrique Vílchez Quesada

Estimados y estimadas estudiantes ¡sean **bienvenidos** al *Taller 5 de resolución de problemas!*

Introducción

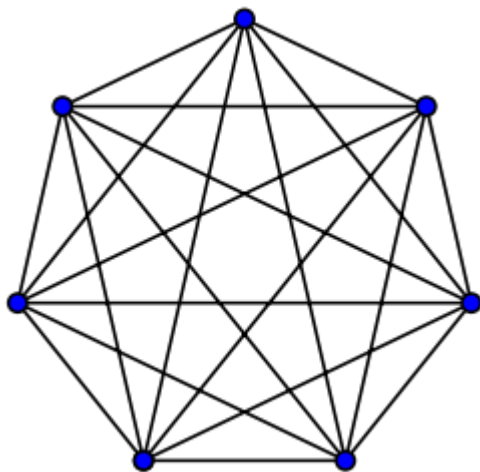
La presente clase está destinada a **estudiar** los **conceptos principales** relacionados con la teoría de grafos **¡Los grafos son más que simples puntos y aristas en el plano!** Si bien es cierto, su representación se fundamenta en este tipo de diagramas **bidimensionales** o **tridimensionales**, las aplicaciones de los grafos reflejan su **diversidad** geométrica derivando en **propiedades matemáticas** interesantes de estudiar.



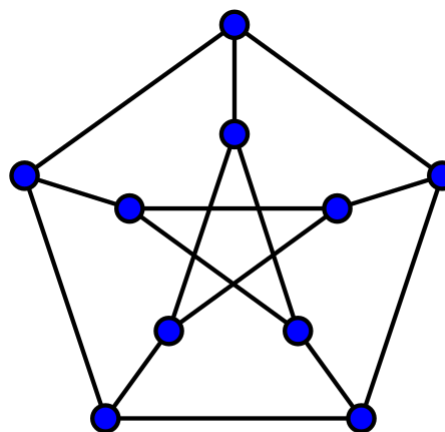
Por ejemplo, la imagen anterior, muestra un grafo sobre un **mapa de regiones** indicando cómo se relacionan entre sí, al asumir los **lados** del grafo como una **carrera** que las une. ¿Será posible pasar una única vez por cada **ciudad**, pasar una única vez por cada **carretera**, conocer **cuántas** trayectorias distintas unen un par de regiones específicas o si pensáramos en añadir a las aristas la distancia entre las ciudades, habrá trayectorias que nos **beneficien más** como viajeros para recorrer menos kilómetros? Preguntas como éstas son las que aborda la teoría de grafos. Les comparto el siguiente **video introductorio** donde de una manera muy amena, el matemático español *Eduardo Sáenz de Cabezón* de la Universidad de La Rioja, nos muestra a la luz de esta teoría: ¿qué tienen en común *Andrés Iniesta*, *Tyrion Lannister* y los amigos en una red social como *Facebook*?

Video: <https://youtu.be/lp-1rvtRYQg>

Entonces, ¿podemos **jugar** con los grafos? El profesor *Sáenz de Cabezón* ha dejado explícito el mensaje: ¡desde luego que **sí**! Observando las figuras presentadas a continuación, ¿será posible dibujar cada una **sin levantar el lápiz del papel**?:



(a)



(b)

La teoría de grafos, como veremos en el transcurso de esta semana, **justifica** que el grafo (a) **sí** es posible **trazarlo** sin levantar el lápiz del papel, pero, por el contrario, el grafo (b) es **imposible**.

Aunque la teoría de grafos permite resolver **problemas muy divertidos**, conforma un insumo teórico con **aplicaciones muy importantes** para estudiar diversas situaciones de naturaleza científica en muchas áreas con cierto rango de **complejidad** y una de ellas son las **ciencias computacionales**. Lo anterior demanda en primera instancia, estudiar los **conceptos principales** de esta teoría, aspecto en el cual se **centrará** los distintos recorridos de aprendizaje de la presente clase. El **viaje comienza** ... con la invitación de sumergirse de manera **proactiva** en las actividades de la semana dentro del contexto del *Taller 5 de resolución de problemas* del presente curso.

Actividades: Taller 5 de resolución de problemas

Se iniciarán las labores académicas de la semana dando **lectura reflexiva y minuciosa** a la *Unidad Didáctica* del tema que nos ocupa, abarcado desde la página 1 hasta la página 39 de dicho documento. La lectura es de **naturaleza obligatoria**. Existen otros recursos complementarios en esta aula virtual que se sugiere revisar: el libro del profesor *Juan Félix Ávila Herrera* y el texto diseñado por el docente *Eithel Trigueros Rodríguez*.

El grupo se ha dividido de manera **aleatoria** en una serie de **subgrupos de cuatro personas como máximo** (ver la lista de los subgrupos **adjunta** a la clase). Cada subgrupo tendrá la tarea de investigar **teóricamente** y de forma **práctica** los conceptos de: grafo, adyacencia, incidencia, tipos de grafos, valencias, ruta, circuito, circuito y camino de *Euler*, circuito y camino de *Hamilton* y, grafo conexo. Los subgrupos trabajarán de manera **colaborativa** en una *wiki* (ya **habilitada** en el espacio del aula virtual del tema de grafos), la **construcción** de un documento con **dos páginas** de extensión máxima, que **resuma** las definiciones y aplicaciones de los términos anteriormente citados, además de incorporar **explicaciones** sobre el uso del software *Wolfram Mathematica* en la **creación** de grafos y **resolución** de circuitos, circuitos y rutas de *Euler*, circuitos y rutas de *Hamilton* y cálculo de valencias. El plazo para la realización de esta tarea es de **cinco días** y se entregará formalmente el documento a más tardar el **día viernes** por un miembro del subgrupo (el **coordinador**) mediante un **enlace** destinado para ello. El **nombre** del archivo enviado será: *Número de subgrupo_Conceptos*.

El día sábado se ¡gestará un **encuentro** en el metaverso **Second Life!**, con la intención de dotar un espacio de **realidad virtual** donde cada subgrupo exponga las ideas principales del documento diseñado de forma colaborativa. El punto de reunión será: <http://maps.secondlife.com/secondlife/Exploratorium/170/181/26>, correspondiente a **Exploratorium**, un recurso para **explorar conceptos matemáticos** y de la **ciencia** en general. Se **debatirá** en tiempo real las ideas de cada uno de los subgrupos, el profesor tomará un rol de **moderador** dando la palabra, ordenando las secuencias y brindado al final, un **resumen** con todo el contenido expuesto. Se invitará a los estudiantes que así lo deseen, acompañar al docente a la galería **Imaginary** ubicada en **Exploratorium**, que consiste en una **réplica** de una galería con ese mismo nombre, desarrollada por el *Instituto Matemático de Oberwolfach*, ubicado en Alemania.



En las **clases presenciales** se abordarán los **contenidos teóricos** para reforzar las ideas más importantes y se propondrán ejercicios de resolución a través del uso del software *Wolfram Mathematica* hacia la **construcción de grafos** y **determinación de trayectorias y circuitos**. El alumno debe realizar **investigaciones previas** sobre el uso del software que garanticen su adecuado **desempeño** en clase. Con este objetivo se recomienda **consultar**: <https://www.symboloo.com/mix/vilcretasgrafos>.

El día miércoles se realizará una **videoconferencia** de consultas cuya participación es de **naturaleza optativa**. El docente aprovechará este espacio para brindar algunas **directrices generales** sobre la elaboración del documento colaborativo.

Finalmente, se ha **habilitado un foro** donde se pretende compartir **aplicaciones** de la teoría de grafos sobre la ingeniería en sistemas de información o ciencias computacionales, además, de otras áreas de carácter científico. Es fundamental participar en el foro desde el **primer día** de la semana (lunes) con el objetivo de garantizar una **reciprocidad** oportuna y apropiada en las interacciones. Hay que responder de manera **puntual** a lo solicitado y **revisar** la redacción de las ideas y la ortografía antes de una publicación. **No es válido** incluir una participación mediante un archivo adjunto.

Los criterios de evaluación se sustentan en valorar el **trabajo colaborativo** sobre los principales conceptos de la teoría de grafos en el documento diseñado y su defensa dentro del metaverso **Second Life**. En general, como se dejó claro en la *Guía Didáctica* del presente curso, el taller será calificado empleando la **rúbrica** correspondiente a la evaluación de los “*talleres de resolución de problemas*”, a saber:

- **Creación crítica y reflexiva de las soluciones:** los estudiantes resuelven de manera creativa e inédita el documento colaborativo.
- **Uso de recursos y software especializado:** los alumnos utilizan recursos didácticos y software como mecanismos de construcción y consulta.
- **Complejidad del contenido:** el documento colaborativo es completo y da una respuesta creativa y puntual.
- **Organización y secuencia:** los estudiantes presentan de forma organizada el contenido, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
- **Originalidad:** el texto colaborativo es original, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.

- **Uso del tiempo:** el documento colaborativo se realiza en el tiempo dispuesto por el docente y la defensa en el metaverso *Second Life* respeta los tiempos asignados por el moderador.
- **Calidad de las participaciones en la wiki:** las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.
- **Participación en el foro:** los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio de los temas y una intervención constructiva.
- **Planificación previa y trabajo en clase:** los estudiantes planifican previamente junto al profesor el diseño y desarrollo del documento colaborativo. El trabajo en clase resuelto por los alumnos tiene un buen desempeño.
- **Dominio en el uso del software *Wolfram Mathematica*:** los alumnos demuestran dominio en el empleo del software *Wolfram Mathematica*, específicamente en la construcción de grafos y resolución de circuitos, circuitos y rutas de *Euler*, circuitos y rutas de *Hamilton* y cálculo de valencias.



Foro

Este espacio tiene como propósito **analizar** las **aplicaciones** de la teoría de grafos en la ingeniería en sistemas de información o ciencias computacionales, además, de otras áreas de carácter científico. Investigue dichas aplicaciones y comparta sus conclusiones **mostrando al menos un ejemplo particular**. No es válido **repetir** aplicaciones ya planteadas por otros compañeros ¡Los espero con sus **valiosos aportes** hasta **finalizar** la semana en marcha!



Clase 2: La molécula de *BUCKYBALL* y los grafos



Enrique Vélchez Quesada

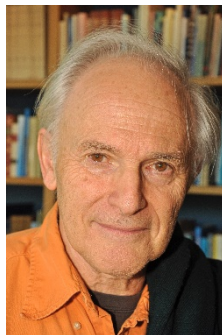
Bienvenidos a esta clase de teoría de grafos donde ¡pondremos en marcha el *Taller 6 de resolución de problemas!*

Introducción

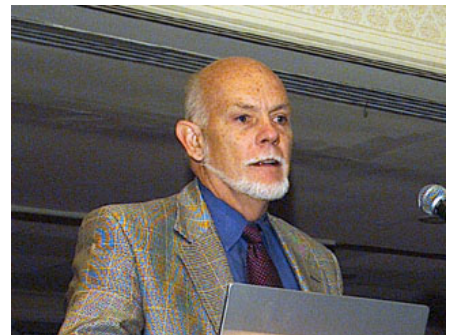
En **1985** los científicos *Robert Curl*, *Harold Kroto* y *Richard Smalley* descubrieron una molécula de carbono de **forma esférica** similar a un balón de fútbol llama molécula *C60* o molécula *BUCKYBALL*. Su descubrimiento les valió el **¡premio nobel** en química! en el año de 1996.



Robert Curl científico
norteamericano



Harold Kroto científico
británico



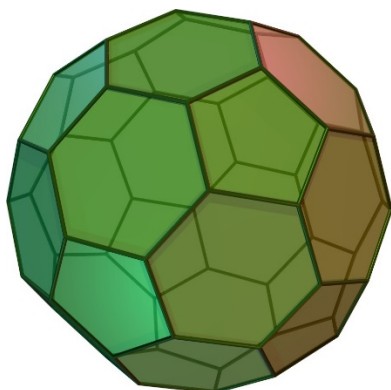
Richard Smalley científico
estadounidense

Este importante aporte científico del compuesto *C60* inició una serie de estudios de **otras moléculas** pertenecientes a la misma **familia** llamada *fullerenos*. Desde la década de los 90 los *fullerenos* han generado mucho interés propiciando una **revolución industrial**

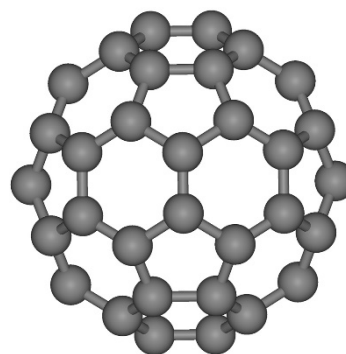
con la intención de construir materiales con este tipo de estructura molecular, al ser **100 veces más resistentes** que el acero. *Richard Smalley* nos lo explica:

Video: <https://youtu.be/mOzijWeC8tY>

La molécula *BUCKYBALL* consiste matemáticamente en un **icosaedro truncado** con 60 vértices y 32 caras.



Icosaedro truncado



Molécula *BUCKYBALL*

En el año 2010 la *NASA* anunció el **descubrimiento** de *fullerenos* en el espacio ¡Un **objeto matemático** por defecto! Puede que no podamos encontrar una raíz cuadrada **caminando** por la calle, pero sí otros tipos de entidades matemáticas que constantemente **aparecen** en la naturaleza y un ejemplo de ello, lo constituye la molécula *BUCKYBALL*.

Pues bien ... en esta clase **dedicaremos nuestra atención** en estudiar las **propiedades matemáticas** de la molécula *BUCKYBALL* y la relación que podemos dar a su estructura con el **tema de los grafos**, puntualmente: los circuitos y rutas de *Euler*, los circuitos y rutas de *Hamilton* y las formas de representación bidimensional de un grafo mediante las matrices de adyacencia e incidencia. Las actividades descritas a continuación entran en correspondencia con el *Taller 6 de resolución de problemas* integrado dentro de las experiencias de aprendizaje de esta materia.

Actividades: Taller 6 de resolución de problemas

Se iniciará las labores dando **lectura** en el módulo de la unidad a los subtemas de: *Representaciones para un grafo* y *Circuitos en un grafo*. Además de ello, es esencial contextualizar el concepto de molécula *BUCKYBALL* y *fullerenos* para lo cual en el aula virtual se han habilitado dos artículos titulados: *La estructura del fullereno C60 y sus aplicaciones* y *Las asombrosas estructuras del carbono: fullerenos, grafenos y nanotubos*. La revisión minuciosa de todos estos materiales es de **naturaleza obligatoria**.

Posteriormente, ya habiendo **analizado** en el material de consulta la terminología principal de la clase, se ejecutará una interesante actividad de carácter **exploratorio**. El objetivo consiste en **investigar** de manera **individual** qué es la molécula *BUCKYBALL* y cuáles son sus principales características matemáticas. Con esta intención se empleará la **aplicación** de realidad aumentada **Augment** (<http://www.augment.com/es>), específicamente en: *Education/Bucky-Ball*, donde es posible visualizar la molécula. A través de este proceso por descubrimiento, se debe responder a las siguientes preguntas: ¿qué es la molécula *BUCKYBALL*?, ¿las caras de este poliedro qué figuras forman?, ¿cuántas aristas contiene?, ¿la molécula podría considerarse un grafo?, si es así, ¿cuáles son sus características: número de nodos y lados, cálculo de valencias, entre otras?, ¿cómo se lograría encontrar el número de aristas mediante el uso de una fórmula?

Los alumnos conformarán de **manera libre** grupos de a lo sumo **cuatro personas** el día lunes de inicio de semana e informarán al docente a través de la **mensajería interna** (en esa misma fecha) los miembros de cada subgrupo y la designación de un coordinador. Mediante una *wiki*, los estudiantes utilizarán los datos recopilados para **construir** a través del uso del software *Wolfram Mathematica*, la molécula *BUCKYBALL*. Con este propósito se sugiere recurrir al comando Grafo perteneciente al paquete de software VilCretas:



Empleando los teoremas y algoritmos de la clase y, el programa *Wolfram Mathematica* los estudiantes indagarán sobre la molécula *BUCKYBALL*, la **existencia** de una ruta y un circuito de *Euler* y de un camino y un circuito de *Hamilton*, dando respuesta desde un punto de vista tanto teórico como práctico. Si es posible alguna de estas trayectorias se deberá dar **un ejemplo**. Asimismo, representarán con al **menos** una matriz de adyacencia y una matriz de incidencia la molécula *BUCKYBALL* ¿Cuántas matrices de adyacencia diferentes representan esta molécula?, ¿cuántas matrices de incidencia distintas? El trabajo colaborativo tendrá una duración de **una semana** dejando todas las huellas de su evolución en la *wiki*, específicamente: indagaciones, formas de razonamiento, discusión crítica, construcción conjunta de resultados y de acuerdos. El coordinador de grupo **adjuntará** el archivo de **extensión .nb** (formato del software *Wolfram Mathematica*) con las soluciones correspondientes, en el espacio de tarea asignado, al finalizar la semana el **día domingo**. El **nombre** del archivo enviado será: *Número de subgrupo_ BUCKYBALL*.

En la clase presencial el profesor **asistirá** a cada subgrupo en la aclaración de dudas teóricas, prácticas o técnicas asociadas con el proceso llevado a cabo. Será **obligatorio** mostrar un **avance** al docente sobre las tareas hasta esa fecha efectuadas, pudiendo demostrar **dominio** de los temas y **consistencia** en las formas de resolución.

En un foro ya habilitado se **ampliarán** los conocimientos base y aplicaciones vinculadas con la molécula *BUCKYBALL* y los *fullerenos* en general. Cada alumno deberá **seleccionar** y **compartir** un recurso de la *Web* que profundice el tema, explicando el por qué de su elección. Participar en el foro con una antelación adecuada garantiza un **aprovechamiento** oportuno. Se responderá de manera **puntual** a lo solicitado en la consigna del foro y se **revisará** en detalle la redacción de las ideas y la ortografía antes

de cualquier publicación. **No es válido** incluir una participación mediante un archivo adjunto.

Los criterios de evaluación de las actividades expuestas con anterioridad se fundamentan en la rúbrica de valoración de los “*talleres de resolución de problemas*”, ya establecida dentro de la *Guía Didáctica* de este curso. Particularmente estos criterios son:

- **Creación crítica y reflexiva de las soluciones:** los estudiantes resuelven de manera creativa e inédita la solución a las distintas preguntas arrojadas con respecto a la molécula *BUCKYBALL* y su relación con la teoría de grafos.
- **Uso de recursos y software especializado:** los alumnos utilizan recursos didácticos y software como mecanismos de construcción, consulta de hipótesis y resolución.
- **Completitud del contenido:** el archivo colaborativo del software *Wolfram Mathematica* es completo y da una respuesta creativa y puntual a las diferentes situaciones planteadas.
- **Organización y secuencia:** los estudiantes presentan de forma organizada el contenido, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
- **Originalidad:** las soluciones construidas con apoyo de software son originales, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.
- **Uso del tiempo:** la conformación de los subgrupos y el archivo *.nb* del programa *Wolfram Mathematica* se realizan en el tiempo dispuesto por el docente.
- **Calidad de las participaciones en la wiki:** las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.
- **Participación en el foro:** los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio y una intervención constructiva.

- **Planificación previa y trabajo en clase:** los estudiantes planifican previamente junto al profesor el diseño y desarrollo de las soluciones integradas en el archivo *.nb*.
- **Dominio en el uso del software *Wolfram Mathematica*:** los alumnos demuestran dominio en el empleo del programa *Wolfram Mathematica* y del paquete de software *VilCretas*.



Foro

Este foro pretende **profundizar** el análisis de la molécula *BUCKYBALL*. Cada participante para ello seleccionará un recurso de la *Web* (imagen, nube de palabras, video, audio, presentación) que brinde un **aporte adicional** sobre la comprensión y aplicaciones de la molécula *BUCKYBALL* y los *fullerenos* en general. No se permitirán recursos **repetidos**. Asimismo, se deberá **justificar** en este espacio la razón de la selección del recurso, especificando los motivos teóricos, prácticos y aplicados que hacen de él un medio de profundización de estos contenidos. Será valioso también, buscar recursos que **vinculen** el tema de los *fullerenos* con la teoría de grafos. El recurso se compartirá a través de la dirección *URL* respectiva o insertando el objeto directamente en la entrada del foro. Con **emoción** ansío leer y visualizar sus aportes. Esta tarea deberá ser realizada en un plazo máximo de **una semana** comenzando el día lunes y finalizando el día domingo ¡Se apreciará la **interacción** en sus comentarios y una discusión **proactiva**!



Clase 3: Los sociogramas una aplicación contemporánea de la teoría de grafos



Enrique Vélchez Quesada

Estimados alumnos y estimadas alumnas es un **placer saludarlos** en este nuevo espacio de clase donde nos enfocaremos en ¡atender el *Taller 6 de resolución de problemas!*

Introducción

Los grafos tienen **muchas aplicaciones** en la vida real y una de ellas consiste en representar las **conexiones generadas** dentro de esta una **red social**. Probablemente muchos de nosotros en **nuestra cotidianidad** empleamos este tipo de recursos con fines **lúdicos** o **académicos**. Un grafo que representa estas conexiones se denomina: **¡sociograma!** La nube de palabras compartida reúne una serie de términos asociados con los sociogramas:



Un sociograma pretende, por lo tanto, mostrar a través de un grafo las **relaciones existentes** dentro de un **espacio social**. La técnica fue desarrollada en los años 30 por *Jacob Levy Moreno* psiquiatra norteamericano, quien estaba interesado en crear una herramienta diagnóstica sobre el **bienestar psicológico** en grupos sociales. Recientemente la teoría de sociogramas ha tomado mucha vigencia por el **uso masivo** de las redes sociales en distintos ámbitos culturales y sus implicaciones de **naturaleza comercial**. Para una empresa puede resultar de vital importancia ¡conocer los **grados de influencia** y **preferencias** de un futuro comprador de sus productos! El video compartido a continuación introduce el **concepto** de sociograma:

Video: <https://youtu.be/SoAh1BB9918>

En la actualidad existen **múltiples programas de cómputo** que facilitan el análisis de la información dentro de un sociograma, a través del empleo de distintos tipos de **algoritmos** y de **métricas**. Ya no es necesario ser ¡un experto en la materia!, dado que estas aplicaciones, muchas de ellas disponibles en la *Web*, permiten aplicar de manera automática los **procedimientos analíticos**. El uso de software en este sentido se torna en una necesidad, pues los grafos de sociogramas usualmente están constituidos por un **número significativo** de nodos y aristas lo cual dificulta su interpretación de manera **intuitiva**. El video próximo, a manera de ejemplo, comparte la utilidad de una herramienta de software con estas características, denominada *NVivo*:

Video: <https://youtu.be/wwwXC6IDK98>

También, el siguiente blog muestra otras interesantes aplicaciones de análisis de redes sociales: <http://ars-uns.blogspot.com/2016/06/30-paquetes-de-software-de-ars.html>. ¡Inclusive! el software *Wolfram Mathematica* **favorece** el estudio de las redes sociales por medio del uso de una serie de **comandos** ya incorporados en el programa. El video compartido a continuación, explica el beneficio de estas prestaciones, mediante una ponencia presentada en el año 2014, dentro del seno de la *Wolfram Technology*

Conference, un **evento académico** anual, organizado por la empresa *Wolfram Research*:

Video: <https://youtu.be/7zZjLBtHf7c>

Esta semana nos abocaremos a estudiar la teoría de grafos como un **recurso** para **analizar sociogramas** nutridos mediante datos recopilados de ¡nuestras propias redes sociales! Como el último *taller de resolución de problemas* del tema de grafos, las actividades de aprendizaje expuestas integran los contenidos estudiados en las semanas anteriores, incluyendo, además, dos nuevos tópicos de interés: el problema del **agente viajero** y el **algoritmo de Dijkstra**.

Actividades Taller 7 de resolución de problemas

El inicio de las actividades toma como punto de partida la lectura obligatoria de la **sección Algoritmo del camino más corto** ubicada en la página 72 de la *Unidad Didáctica*. De forma complementaria se pueden revisar los materiales: el libro del profesor *Juan Félix Ávila Herrera* y el texto de autoría del docente *Eithel Trigueros Rodríguez*. Con la intención de **contextualizar** los temas de sociogramas y el análisis de redes sociales, es esencial la lectura obligatoria de los recursos: *El análisis de las interacciones grupales: las aplicaciones SOCIOS* e *Introducción al Análisis de Redes Sociales*. De forma optativa se sugiere la lectura: *El sociograma*.

El docente dividirá al grupo en subgrupos al **azar** con un máximo de **cuatro integrantes**. En una *wiki* se analizarán cuáles son las **conexiones digitales** existentes entre los miembros del subgrupo, sea en *Facebook*, *Twitter* u otra red social de sugestión común, respondiendo a la pregunta: ¿qué conexiones **poseen entre sí** que consideren relevantes en su vida social en la *Internet*? Con la información recopilada y discutida los alumnos construirán un **grafo social** donde cada uno constituya un nodo. Se podrán **añadir nodos** que representen otras personas que no estén en el subgrupo, páginas *Web*, intereses, juegos o cualquier otro aspecto de relevancia. El grafo deberá reflejar **todas las conexiones posibles** entre los **participantes** por lo que se espera posea una **cantidad** de nodos y aristas relativamente **significativa**.

Utilizando el software *Wolfram Mathematica* y la instrucción Grafo del paquete VilCretas, los estudiantes construirán un **grafo social 3D** con la información recolectada en el paso anterior. **Clasificarán** el grafo construido en alguno de los tipos vistos en clase: completo, bipartito, mariposa, rueda, estrella, círculo, camino, regular, red, entre otros. Usando los grados o valencias **determinarán** la cantidad de aristas que lo caracterizan. Con el grafo diseñado, se **responderán** las siguientes preguntas **justificando** teóricamente y a través del uso del paquete VilCretas como recurso de apoyo: ¿posee circuitos de *Euler*? y ¿posee circuitos de *Hamilton*?

Los estudiantes **reconstruirán** el grafo creado asignando **pesos** a cada una de las aristas en función del **nivel de importancia** en las relaciones sociales entre los nodos. Dos nodos débiles pero vinculados con una arista, podrían tener un peso 1, por el contrario, dos nodos con una relación fuerte podrían tener asociado un valor de 100, las escalas **serán definidas por cada subgrupo**, explicando en detalle el criterio utilizado para ello. Posteriormente, se seleccionarán **dos nodos no adyacentes** en el grafo social, con el objetivo de **aplicar** sobre ellos el algoritmo de la longitud del camino más corto, para determinar una **ruta de longitud más larga** usando el paquete VilCretas y el software *Wolfram Mathematica* ¿Qué representa el peso máximo de la ruta? Se sugiere consultar el siguiente video explicativo, sobre el empleo de un valioso comando, llamado **ALDijkstra**:

Video: <https://youtu.be/8OKGR7nK3V8>

Finalmente, los estudiantes **responderán** a la pregunta: ¿es posible resolver el problema del agente viajero (*PAV*) en el grafo creado?, si es así, se deberá **devolver** dicha solución y en caso contrario, **modificar** el grafo con la intención de poder dar respuesta al *PAV*, explicando el criterio de alteración en el grafo original. En cualquier caso, se **animará visualmente** la solución utilizando la instrucción AnimarGrafo. Para **más detalles** sobre el uso de este comando se puede consultar:

Video: <https://youtu.be/T4tl4bwCJbl>

Se solicitará a cada subgrupo la **elaboración de un video** que explique la presentación de su grafo social tridimensional y el proceso realizado en cada una de las **etapas** ya descritas. El video será publicado en: *YouTube* o *Vimeo*, con una duración máxima de cinco minutos y mínima de cuatro. El video deberá tener una **buena calidad** visual y de audio e integrar un guion **llamativo** y **original**. El enlace a la producción audiovisual se entregará por medio de un espacio de tarea al finalizar la semana el día domingo, por parte del **coordinador** de grupo.

Todo el proceso requerido para llevar a **buen término** las actividades ya compartidas, deberá quedar **consignado** en su totalidad en la *wiki* de trabajo, mostrando los distintos avances, formas de resolución, dominio de los contenidos y construcción conjunta de tesis. Las clases presenciales servirán como insumo para la **aclaración de dudas** y **defensa** de las propuestas al responder a las consultas del docente respecto a los **avances**. Los alumnos tendrán la responsabilidad de **demostrar** un adecuado desempeño teórico, práctico y técnico. El día miércoles se realizará en esta aula virtual un **chat** de consultas cuya participación es de **naturaleza optativa**.

Finalmente, en el foro vinculado a la clase se investigará de manera individual distintas herramientas para el análisis de sociogramas y redes sociales, dando **amplitud** a los temas tratados. Se insta a los estudiantes a participar en el foro desde el **primer día** de la semana (lunes) con el objetivo de garantizar **interacciones** oportunas y apropiadas. Se dará una respuesta **puntual** a lo solicitado y se **revisará** minuciosamente la redacción de las ideas y la ortografía. **No es válido** incluir una participación mediante un archivo adjunto.

La evaluación del *Taller 7 de resolución de problemas* se sustenta en la rúbrica desarrollada dentro de la *Guía Didáctica* del presente curso. Los criterios son:

- **Creación crítica y reflexiva de las soluciones:** los estudiantes resuelven de manera creativa e inédita las soluciones a los problemas propuestos y, el diseño y desarrollo del video.

- **Uso de recursos y software especializado:** los alumnos utilizan recursos didácticos y software como mecanismos de construcción y consulta.
- **Compleitud del contenido:** el video elaborado y las formas de resolución en la *wiki* son completas y dan una respuesta creativa y puntual.
- **Organización y secuencia:** los estudiantes presentan de forma organizada el contenido en el video y la *wiki*, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
- **Originalidad:** el video es original, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.
- **Uso del tiempo:** el avance del trabajo colaborativo en la *wiki* y el video se realizan en el tiempo dispuesto por el docente.
- **Calidad de las participaciones en la *wiki*:** las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.
- **Participación en el foro:** los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio de los temas y una intervención constructiva.
- **Planificación previa y trabajo en clase:** los estudiantes planifican previamente junto al profesor las soluciones a los problemas propuestos y, el diseño y desarrollo del video. El trabajo en clase resuelto por los alumnos tiene un buen desempeño.
- **Dominio en el uso del software *Wolfram Mathematica*:** los alumnos demuestran dominio en el empleo del software *Wolfram Mathematica* y el paquete VilCretas.



Foro

Este foro tiene como objetivo **estudiar** herramientas de análisis de sociogramas. Se iniciará indagando en la *Web* distintas **herramientas** disponibles para el **análisis** de sociogramas y redes sociales. Realicen pruebas, sean creativos, utilicen sus propias redes sociales para ¡experimentar! Como sugerencia se les comparte este interesante

blog: <http://ars-uns.blogspot.com/2016/06/30-paquetes-de-software-de-ars.html>.

Seleccionen al menos **tres herramientas** y publiquen en este espacio una **justificación** teóricamente fundamentada de sus elecciones y algunos *screen shots* como evidencia del trabajo exploratorio realizado al **utilizar** cada software. La tarea deberá ser concretada en un plazo de **una semana** comenzando el día lunes y finalizando el día domingo ¡Se valorará la **interacción** entre las participaciones y el **uso creativo** de las herramientas de análisis!



5. Captura de pantalla de las clases

Clase 1: Los grafos ... más allá de un conjunto de puntos y aristas

- Bienvenida:

Clase: Taller 5 de resolución de problemas

Clase: Los grafos ... más allá de un conjunto de puntos y aristas



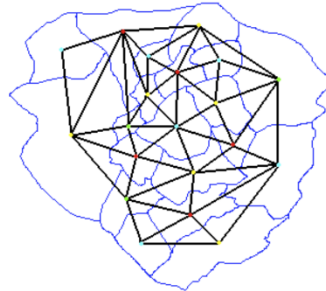
Enrique Vilchez Quesada

Estimados y estimadas estudiantes izean **bienvenidos** al *Taller 5 de resolución de problemas!*

Introducción:

Introducción

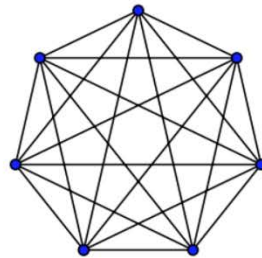
La presente clase está destinada a **estudiar** los **conceptos principales** relacionados con la teoría de grafos **¡Los grafos son más que simples puntos y aristas en el plano!** Si bien es cierto, su representación se fundamenta en este tipo de diagramas **bidimensionales** o **tridimensionales**, las aplicaciones de los grafos reflejan su **diversidad** geométrica derivando en **propiedades matemáticas** interesantes de estudiar.



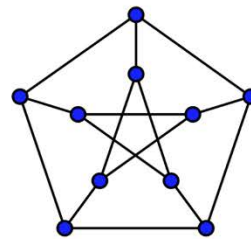
Por ejemplo, la imagen anterior, muestra un grafo sobre un **mapa de regiones** indicando cómo se relacionan entre sí, al asumir los **lados** del grafo como una **carrera** que las une. ¿Será posible pasar una única vez por cada **ciudad**, pasar una única vez por cada **carretera**, conocer **cuántas** trayectorias distintas unen un par de regiones específicas o si pensáramos en añadir a las aristas la distancia entre las ciudades, habrá trayectorias que nos **beneficien más** como viajeros para recorrer menos kilómetros? Preguntas como éstas son las que aborda la teoría de grafos. Les comparto el siguiente **video introductorio** donde de una manera muy amena, el matemático español *Eduardo Sáenz de Cabezón* de la Universidad de La Rioja, nos muestra a la luz de esta teoría: ¿qué tienen en común *Andrés Iniesta*, *Tyrion Lannister* y los amigos en una red social como *Facebook*?



Entonces, ¿podemos **jugar** con los grafos? El profesor *Sáenz de Cabezón* ha dejado explícito el mensaje: ¡desde luego que **sí!** Observando las figuras presentadas a continuación, ¿será posible dibujar cada una **sin levantar el lápiz del papel**?:



(a)



(b)

La teoría de grafos, como veremos en el transcurso de esta semana, **justifica** que el grafo (a) **sí** es posible **trazarlo** sin levantar el lápiz del papel, pero, por el contrario, el grafo (b) es **imposible**.

Aunque la teoría de grafos permite resolver **problemas muy divertidos**, conforma un insumo teórico con **aplicaciones muy importantes** para estudiar diversas situaciones de naturaleza científica en muchas áreas con cierto rango de **complejidad** y una de ellas son las **ciencias computacionales**. Lo anterior demanda en primera instancia, estudiar los **conceptos principales** de esta teoría, aspecto en el cual se **centrará** los distintos recorridos de aprendizaje de la presente clase. El **viaje comienza** ... con la invitación de sumergirse de manera **proactiva** en las actividades de la semana dentro del contexto del *Taller 5 de resolución de problemas* del presente curso.

■ Actividades:

Actividades: *Taller 5 de resolución de problemas*

Se iniciarán las labores académicas de la semana dando **lectura reflexiva y minuciosa** a la *Unidad Didáctica* del tema que nos ocupa, abarcado desde la página 1 hasta la página 39 de dicho documento. La lectura es de **naturaleza obligatoria**. Existen otros recursos complementarios en esta aula virtual que se sugiere revisar: el **libro del profesor Juan Félix Ávila Herrera** y el texto diseñado por el **docente Eithel Trigueros Rodríguez**.

El grupo se ha dividido de manera **aleatoria** en una serie de **subgrupos de cuatro personas como máximo** (ver la lista de los subgrupos **adjunta** a la clase). Cada subgrupo tendrá la tarea de investigar **teóricamente** y de forma **práctica** los conceptos de: grafo, adyacencia, incidencia, tipos de grafos, valencias, ruta, circuito, camino de *Euler*, circuito y camino de *Hamilton* y, grafo conexo. Los subgrupos trabajarán de manera **colaborativa** en una *wiki* (ya **habilitada** en el espacio del aula virtual del tema de grafos), la **construcción** de un documento con **dos páginas** de extensión máxima, que **resume** las definiciones y aplicaciones de los términos anteriormente citados, además de incorporar **explicaciones** sobre el uso del software *Wolfram Mathematica* en la **creación** de grafos y **resolución** de circuitos, circuitos y rutas de *Euler*, circuitos y rutas de *Hamilton* y cálculo de valencias. El plazo para la realización de esta tarea es de **cinco días** y se entregará formalmente el documento a más tardar el **día viernes** por un miembro del subgrupo (el **coordinador**) mediante un **enlace** destinado para ello. El **nombre** del archivo enviado será: *Número de subgrupo_Conceptos*.

El día sábado se **igstará** un **encuentro** en el metaverso **Second Life!**, con la intención de dotar un espacio de **realidad virtual** donde cada subgrupo exponga las ideas principales del documento diseñado de forma colaborativa. El punto de reunión será:

<http://maps.secondlife.com/secondlife/Exploratorium/170/181/26>, correspondiente a **Exploratorium**, un recurso para **explorar conceptos matemáticos** y de la **ciencia** en general. Se **debatirá** en tiempo real las ideas de cada uno de los subgrupos, el profesor tomará un rol de **moderador** dando la palabra, ordenando las secuencias y brindado al final, un **resumen** con todo el contenido expuesto. Se invitará a los estudiantes que así lo deseen, acompañar al docente a la galería **Imaginary** ubicada en **Exploratorium**, que consiste en una **réplica** de una galería con ese mismo nombre, desarrollada por el *Instituto Matemático de Oberwolfach*, ubicado en Alemania.



En las **clases presenciales** se abordarán los **contenidos teóricos** para reforzar las ideas más importantes y se propondrán ejercicios de resolución a través del uso del software *Wolfram Mathematica* hacia la **construcción de grafos** y **determinación de trayectorias y circuitos**. El alumno debe realizar **investigaciones previas** sobre el uso del software que garanticen su adecuado **desempeño** en clase. Con este objetivo se recomienda **consultar**: <https://www.symbaloo.com/mix/vilcretasgrafos>.

El día miércoles se realizará una **videoconferencia** de consultas cuya participación es de **naturaleza optativa**. El docente aprovechará este espacio para brindar algunas **directrices generales** sobre la elaboración del documento colaborativo.

Finalmente, se ha **habilitado un foro** donde se pretende compartir **aplicaciones** de la teoría de grafos sobre la ingeniería en sistemas de información o ciencias computacionales, además, de otras áreas de carácter científico. Es fundamental participar en el foro desde el **primer día** de la semana (lunes) con el objetivo de garantizar una **reciprocidad** oportuna y apropiada en las interacciones. Hay que responder de manera **puntual** a lo solicitado y **revisar** la redacción de las ideas y la ortografía antes de una publicación. **No es válido** incluir una participación mediante un archivo adjunto.

Los criterios de evaluación se sustentan en valorar el **trabajo colaborativo** sobre los principales conceptos de la teoría de grafos en el documento diseñado y su defensa dentro del metaverso **Second Life**. En general, como se dejó claro en la *Guía Didáctica* del presente curso, el taller será calificado empleando la **rúbrica** correspondiente a la evaluación de los "talleres de resolución de problemas", a saber:

- **Creación crítica y reflexiva de las soluciones:** los estudiantes resuelven de manera creativa e inédita el documento colaborativo.
- **Uso de recursos y software especializado:** los alumnos utilizan recursos didácticos y software como mecanismos de construcción y consulta.
- **Complejidad del contenido:** el documento colaborativo es completo y da una respuesta creativa y puntual.
- **Organización y secuencia:** los estudiantes presentan de forma organizada el contenido, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
- **Originalidad:** el texto colaborativo es original, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.
- **Uso del tiempo:** el documento colaborativo se realiza en el tiempo dispuesto por el docente y la defensa en el metaverso *Second Life* respeta los tiempos asignados por el moderador.
- **Calidad de las participaciones en la wiki:** las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.
- **Participación en el foro:** los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio de los temas y una intervención constructiva.
- **Planificación previa y trabajo en clase:** los estudiantes planifican previamente junto al profesor el diseño y desarrollo del documento colaborativo. El trabajo en clase resuelto por los alumnos tiene un buen desempeño.
- **Dominio en el uso del software Wolfram Mathematica:** los alumnos demuestran dominio en el empleo del software *Wolfram Mathematica*, específicamente en la construcción de grafos y resolución de circuitos, circuitos y rutas de *Euler*, circuitos y rutas de *Hamilton* y cálculo de valencias.



Wiki:

Subgrupo 1-Wiki: Taller 5 de resolución de problemas

En este espacio de **trabajo colaborativo** investigarán **teóricamente** y de **forma práctica** los conceptos de: grafo, adyacencia, incidencia, tipos de grafos, valencias, ruta, circuito, camino de *Euler*, circuito y camino de *Hamilton* y, grafo conexo.

Ver Editar Comentarios Historia Mapa Ficheros Administración

Grupos separados: Todos los participantes

Versión imprimible

Trabajo colaborativo

Investigación teórica

Esta página esta destinada a realizar el trabajo colaborativo de la semana, imanos a la obra!

Espacio de entrega de tarea:

Espacio de entrega: Documento colaborativo

En este espacio se entregará el documento con **dos páginas** de extensión máxima, que **resuma** las definiciones y aplicaciones de los términos introductorios de la teoría de grafos, además de incorporar **explicaciones** sobre el uso del software *Wolfram Mathematica* en la **creación** de grafos y **resolución** de circuitos, circuitos y rutas de *Euler*, circuitos y rutas de *Hamilton* y cálculo de valencias. El **nombre** del archivo enviado será: *Número de subgrupo_Conceptos*.

Sumario de calificaciones

Participantes	0
Enviados	0
Pendientes por calificar	0

Ver/Calificar todas las entregas

Calificación


- Foro:

Foro: Aplicaciones de la teoría de grafos

En este foro se **analizarán** algunas **aplicaciones** de la teoría de grafos.



Añadir un nuevo tema de discusión

Tema	Comenzado por	Réplicas	Último mensaje
Consigna del foro	 ENRIQUE VILCHEZ QUESADA 401650194	0	ENRIQUE VILCHEZ QUESADA 401650194 Tue, 1 de May de 2018, 08:13

Foro: Aplicaciones de la teoría de grafos

Consigna del foro

 Suscrito

Mostrar respuestas anidadas ▾

Mover este tema a... ▾

Mover

Fijar



Consigna del foro

de ENRIQUE VILCHEZ QUESADA 401650194 - Tuesday, 1 de May de 2018, 08:13

Este espacio tiene como propósito **analizar** las **aplicaciones** de la teoría de grafos en la ingeniería en sistemas de información o ciencias computacionales, además, de otras áreas de carácter científico. Investigue dichas aplicaciones y comparta sus conclusiones **mostrando al menos un ejemplo particular**. No es válido **repetir** aplicaciones ya planteadas por otros compañeros ¡Los espero con sus **valiosos aportes** hasta **finalizar** la semana en marcha!



[Enlace permanente](#) | [Editar](#) | [Borrar](#) | [Responder](#)

Clase 2: La molécula de **BUCKYBALL** y los grafos

- Bienvenida:

Clase: Taller 6 de resolución de problemas

Clase: La molécula de **BUCKYBALL** y los grafos



Enrique Vilchez Quesada

Bienvenidos a esta clase de teoría de grafos donde pondremos en marcha el *Taller 6 de resolución de problemas!*

Introducción:

Introducción

En **1985** los científicos *Robert Curl*, *Harold Kroto* y *Richard Smalley* descubrieron una molécula de carbono de **forma esférica** similar a un balón de fútbol llama molécula *C₆₀* o molécula *BUCKYBALL*. Su descubrimiento les valió el **ipremio nobel** en química en el año de 1996.



*Robert
Curl* científico norteamericano



*Harold
Kroto* científico
británico

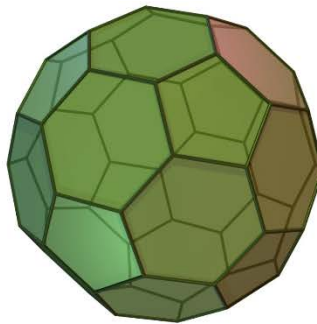


*Richard
Smalley* científico estadounidense

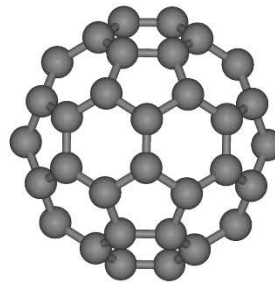
Este importante aporte científico del compuesto *C₆₀* inició una serie de estudios de **otras moléculas** pertenecientes a la misma **familia** llamada *fullerenos*. Desde la década de los 90 los *fullerenos* han generado mucho interés propiciando una **revolución industrial** con la intención de construir materiales con este tipo de estructura molecular, al ser **100 veces más resistentes** que el acero. *Richard Smalley* nos lo explica:



La molécula *BUCKYBALL* consiste matemáticamente en un **icosaedro truncado** con 60 vértices y 32 caras.



Icosaedro truncado



Molécula *BUCKYBALL*

En el año 2010 la *NASA* anunció el **descubrimiento** de *fullerenos* en el espacio ¡Un **objeto matemático** por defecto! Puede que no podamos encontrar una raíz cuadrada **caminando** por la calle, pero sí otros tipos de entidades matemáticas que constantemente **aparecen** en la naturaleza y un ejemplo de ello, lo constituye la molécula *BUCKYBALL*.

Pues bien ... en esta clase **dedicaremos nuestra atención** en estudiar las **propiedades matemáticas** de la molécula *BUCKYBALL* y la relación que podemos dar a su estructura con el **tema de los grafos**, puntualmente: los circuitos y rutas de *Euler*, los circuitos y rutas de *Hamilton* y las formas de representación bidimensional de un grafo mediante las matrices de adyacencia e incidencia. Las actividades descritas a continuación entran en correspondencia con el *Taller 6 de resolución de problemas* integrado dentro de las experiencias de aprendizaje de esta materia.

■ Actividades:

Actividades: Taller 6 de resolución de problemas

Se iniciará las labores dando **lectura** en el **módulo de la unidad** a los subtemas de: *Representaciones para un grafo y Circuitos en un grafo*. Además de ello, es esencial contextualizar el concepto de molécula *BUCKYBALL* y *fullerenos* para lo cual en el aula virtual se han habilitado dos artículos titulados: *La estructura del fullereno C60 y sus aplicaciones* y *Las asombrosas estructuras del carbono: fullerenos, grafenos y nanotubos*. La revisión minuciosa de todos estos materiales es de **naturaleza obligatoria**.

Posteriormente, ya habiendo **analizado** en el material de consulta la terminología principal de la clase, se ejecutará una interesante actividad de carácter **exploratorio**. El objetivo consiste en **investigar** de manera **individual** qué es la molécula *BUCKYBALL* y cuáles son sus principales características matemáticas. Con esta intención se empleará la **aplicación** de realidad aumentada **Augment** (<http://www.augment.com/es>), específicamente en: *Education/Bucky-Ball*, donde es posible visualizar la molécula. A través de este proceso por descubrimiento, se debe responder a las siguientes preguntas: ¿qué es la molécula *BUCKYBALL*?, ¿las caras de este poliedro qué figuras forman?, ¿cuántas aristas contiene?, ¿la molécula podría considerarse un grafo?, si es así, ¿cuáles son sus características: número de nodos y lados, cálculo de valencias, entre otras?, ¿cómo se lograría encontrar el número de aristas mediante el uso de una fórmula?

Los alumnos conformarán de **manera libre** grupos de a lo sumo **cuatro personas** el día lunes de inicio de semana e informarán al docente a través de la **mensajería interna** (en esa misma fecha) los miembros de cada subgrupo y la designación de un coordinador. Mediante una **wiki**, los estudiantes utilizarán los datos recopilados para **construir** a través del uso del software *Wolfram Mathematica*, la molécula *BUCKYBALL*. Con este propósito se sugiere recurrir al comando **Grafo** perteneciente al paquete de software **ViICretas**:



Empleando los teoremas y algoritmos de la clase y, el programa *Wolfram Mathematica* los estudiantes indagarán sobre la molécula *BUCKYBALL*, la **existencia** de una ruta y un circuito de *Euler* y de un camino y un circuito de *Hamilton*, dando respuesta desde un punto de vista tanto teórico como práctico. Si es posible alguna de estas trayectorias se deberá dar **un ejemplo**. Asimismo, representarán con **al menos** una matriz de adyacencia y una matriz de incidencia la molécula *BUCKYBALL* ¿Cuántas matrices de adyacencia diferentes representan esta molécula?, ¿cuántas matrices de incidencia distintas? El trabajo colaborativo tendrá una duración de **una semana** dejando todas las huellas de su evolución en la **wiki**, específicamente: indagaciones, formas de razonamiento, discusión crítica, construcción conjunta de resultados y de acuerdos. El coordinador de grupo **adjuntará** el archivo de **extensión .nb** (formato del software *Wolfram Mathematica*) con las soluciones correspondientes, en el espacio de **tarea asignado**, al finalizar la semana el **día domingo**. El **nombre** del archivo enviado será: *Número de subgrupo_ BUCKYBALL*.

En la clase presencial el profesor **asistirá** a cada subgrupo en la aclaración de dudas teóricas, prácticas o técnicas asociadas con el proceso llevado a cabo. Será **obligatorio** mostrar un **avance** al docente sobre las tareas hasta esa fecha efectuadas, pudiendo demostrar **dominio** de los temas y **consistencia** en las formas de resolución.

En un **foro** ya habilitado se **ampliarán** los conocimientos base y aplicaciones vinculadas con la molécula *BUCKYBALL* y los *fullerenos* en general. Cada alumno deberá **seleccionar** y **compartir** un recurso de la *Web* que profundice el tema, explicando el por qué de su elección. Participar en el **foro** con una antelación adecuada garantiza un **aprovechamiento** oportuno. Se responderá de manera **puntual** a lo solicitado en la consigna del **foro** y se **revisará** en detalle la redacción de las ideas y la ortografía antes de cualquier publicación. **No es válido** incluir una participación mediante un archivo adjunto.

Los criterios de evaluación de las actividades expuestas con anterioridad se fundamentan en la rúbrica de valoración de los "talleres de resolución de problemas", ya establecida dentro de la *Guía Didáctica* de este curso. Particularmente estos criterios son:

- **Creación crítica y reflexiva de las soluciones:** los estudiantes resuelven de manera creativa e inédita la solución a las distintas preguntas arrojadas con respecto a la molécula *BUCKYBALL* y su relación con la teoría de grafos.
- **Uso de recursos y software especializado:** los alumnos utilizan recursos didácticos y software como mecanismos de construcción, consulta de hipótesis y resolución.
- **Complejidad del contenido:** el archivo colaborativo del software *Wolfram Mathematica* es completo y da una respuesta creativa y puntual a las diferentes situaciones planteadas.
- **Organización y secuencia:** los estudiantes presentan de forma organizada el contenido, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
- **Originalidad:** las soluciones construidas con apoyo de software son originales, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.
- **Uso del tiempo:** la conformación de los subgrupos y el archivo *.nb* del programa *Wolfram Mathematica* se realizan en el tiempo dispuesto por el docente.
- **Calidad de las participaciones en la wiki:** las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.
- **Participación en el foro:** los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio y una intervención constructiva.
- **Planificación previa y trabajo en clase:** los estudiantes planifican previamente junto al profesor el diseño y desarrollo de las soluciones integradas en el archivo *.nb*.
- **Dominio en el uso del software Wolfram Mathematica:** los alumnos demuestran dominio en el empleo del programa *Wolfram Mathematica* y del paquete de software *ViICretas*.




- **Wiki:**

Subgrupo 1-Wiki: Taller 6 de resolución de problemas

Investiguen de manera individual **qué es** la molécula *BUCKYBALL* y cuáles son sus **principales características matemáticas**. Con esta intención empleen la aplicación de realidad aumentada *Augment* (<http://www.augment.com/es>), específicamente en: *Education/Bucky-Ball*, donde es posible **visualizar** la molécula. Tomando como base este proceso exploratorio, reviertan en este espacio la **respuesta** a las siguientes preguntas: ¿qué es la molécula *BUCKYBALL*?, ¿las caras de este poliedro qué figuras forman?, ¿cuántas aristas contiene?, ¿la molécula podría considerarse un grafo?, si es así, ¿cuáles son sus características: número de nodos y lados, cálculo de valencias, entre otras?, ¿cómo se lograría encontrar el número de aristas mediante el uso de una fórmula? Utilizando los datos **compartidos** y **recopilados construyan** a través del uso del software *Wolfram Mathematica* la molécula *BUCKYBALL*. Empleando los teoremas y algoritmos de la clase y, el programa *Wolfram Mathematica* indaguen sobre la molécula *BUCKYBALL*, la **existencia** de una ruta y un circuito de *Euler* y de un camino y un circuito de *Hamilton*, dando respuesta desde un punto de vista tanto **teórico como práctico**. Si es posible alguna de estas trayectorias se deberá **dar un ejemplo**. Asimismo, **representarán** con al menos una matriz de adyacencia y una matriz de incidencia la molécula *BUCKYBALL* ¿Cuántas matrices de adyacencia diferentes representan esta molécula?, ¿cuántas matrices de incidencia distintas?

Ver [Editar](#) [Comentarios](#) [Historia](#) [Mapa](#) [Ficheros](#) [Administración](#)

Grupos separados: Todos los participantes

 [Versión imprimible](#)

Trabajo colaborativo

La molécula *BUCKYBALL*

Esta página esta destinada a realizar el trabajo colaborativo semanal, ilos espero con sus aportes!

- **Espacio de entrega de tarea:**

Espacio de entrega: Archivo del software Wolfram Mathematica

En este espacio se entregará el archivo de **extensión .nb** (formato del software *Wolfram Mathematica*) con las soluciones correspondientes a los **problemas planteados**. El **nombre** del archivo enviado será: *Número de subgrupo_ BUCKYBALL*.

Sumario de calificaciones

Participantes	0
Enviados	0
Pendientes por calificar	0

[Ver/Calificar todas las entregas](#)

[Calificación](#)


- **Foro:**

Foro: Análisis de la molécula *BUCKYBALL*

En este espacio se **analizarán** las **características principales** de la molécula *BUCKYBALL*.



[Añadir un nuevo tema de discusión](#)

Tema	Comenzado por	Réplicas	Último mensaje
Consigna del foro	 ENRIQUE VILCHEZ QUESADA 401650194	0	ENRIQUE VILCHEZ QUESADA 401650194 Tue, 1 de May de 2018, 08:15

Consigna del foro

 Suscrito

Mostrar respuestas anidadas ▾

Mover este tema a... ▾

Mover

Fijar



Consigna del foro

de ENRIQUE VILCHEZ QUESADA 401650194 - Tuesday, 1 de May de 2018, 08:15

Este foro pretende **profundizar** el análisis de la molécula *BUCKYBALL*. Cada participante para ello seleccionará un recurso de la *Web* (imagen, nube de palabras, video, audio, presentación) que brinde un **aporte adicional** sobre la comprensión y aplicaciones de la molécula *BUCKYBALL* y los *fullerenos* en general. No se permitirán recursos **repetidos**. Asimismo, se deberá **justificar** en este espacio la razón de la selección del recurso, especificando los motivos teóricos, prácticos y aplicados que hacen de él un medio de profundización de estos contenidos. Será valioso también, buscar recursos que **vinculen** el tema de los *fullerenos* con la teoría de grafos. El recurso se compartirá a través de la dirección *URL* respectiva o insertando el objeto directamente en la entrada del foro. Con **emoción** ansío leer y visualizar sus aportes. Esta tarea deberá ser realizada en un plazo máximo de **una semana** comenzando el día lunes y finalizando el día domingo ¡Se apreciará la **interacción** en sus comentarios y una discusión **proactiva**!



[Enlace permanente](#) | [Editar](#) | [Borrar](#) | [Responder](#)

Clase 3: Los sociogramas una aplicación contemporánea de la teoría de grafos

- Bienvenida:

Clase: Taller 7 de resolución de problemas

Clase: Los sociogramas una aplicación contemporánea de la teoría de grafos



Enrique Vilchez Quesada

Estimados alumnos y estimadas alumnas es un **placer saludarlos** en este nuevo espacio de clase donde nos enfocaremos en atender el *Taller 6 de resolución de problemas*!

Introducción:

Introducción

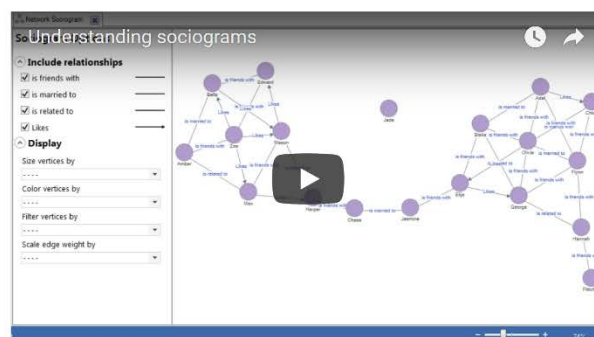
Los grafos tienen **muchas aplicaciones** en la vida real y una de ellas consiste en representar las **conexiones generadas** dentro de esta una **red social**. Probablemente muchos de nosotros en **nuestra cotidianidad** empleamos este tipo de recursos con fines **lúdicos** o **académicos**. Un grafo que representa estas conexiones se denomina: **isociograma!** La nube de palabras compartida reúne una serie de términos asociados con los sociogramas:



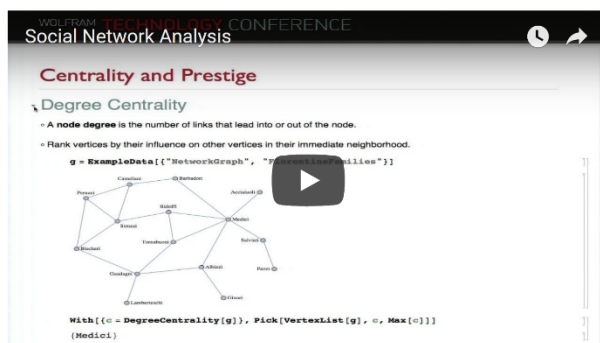
Un sociograma pretende, por lo tanto, mostrar a través de un grafo las **relaciones existentes** dentro de un **espacio social**. La técnica fue desarrollada en los años 30 por **Jacob Levy Moreno** psiquiatra norteamericano, quien estaba interesado en crear una herramienta diagnóstica sobre el **bienestar psicológico** en grupos sociales. Recientemente la teoría de sociogramas ha tomado mucha vigencia por el **uso masivo** de las redes sociales en distintos ámbitos culturales y sus implicaciones de **naturaleza comercial**. Para una empresa puede resultar de vital importancia iconocer los **grados de influencia** y **preferencias** de un futuro comprador de sus productos! El video compartido a continuación introduce el **concepto** de sociograma:



En la actualidad existen **múltiples programas de cómputo** que facilitan el análisis de la información dentro de un sociograma, a través del empleo de distintos tipos de **algoritmos** y de **métricas**. Ya no es necesario ser un experto en la materia!, dado que estas aplicaciones, muchas de ellas disponibles en la **Web**, permiten aplicar de manera automática los **procedimientos analíticos**. El uso de software en este sentido se torna en una necesidad, pues los grafos de sociogramas usualmente están constituidos por un **número significativo** de nodos y aristas lo cual dificulta su interpretación de manera **intuitiva**. El video próximo, a manera de ejemplo, comparte la utilidad de una herramienta de software con estas características, denominada **NVivo**:



También, el siguiente blog muestra otras interesantes aplicaciones de análisis de redes sociales: <http://ars-uns.blogspot.com/2016/06/30-paquetes-de-software-de-ars.html>. ¡Inclusive! el software *Wolfram Mathematica* favorece el estudio de las redes sociales por medio del uso de una serie de **comandos** ya incorporados en el programa. El video compartido a continuación, explica el beneficio de estas prestaciones, mediante una ponencia presentada en el año 2014, dentro del seno de la *Wolfram Technology Conference*, un **evento académico** anual, organizado por la empresa *Wolfram Research*:



Esta semana nos abocaremos a estudiar la teoría de grafos como un **recurso** para **analizar sociogramas** nutridos mediante datos recopilados de inuestras propias redes sociales! Como el último *taller de resolución de problemas* del tema de grafos, las actividades de aprendizaje expuestas integran los contenidos estudiados en las semanas anteriores, incluyendo, además, dos nuevos tópicos de interés: el problema del **agente viajero** y el **algoritmo de Dijkstra**.

■ Actividades:

Actividades Taller 7 de resolución de problemas

El inicio de las actividades toma como punto de partida la lectura obligatoria de la **sección Algoritmo del camino más corto** ubicada en la página 72 de la *Unidad Didáctica*. De forma complementaria se pueden revisar los materiales: el **libro del profesor Juan Félix Ávila Herrera** y el texto de autoría del **docente Eithel Trigueros Rodríguez**. Con la intención de **contextualizar** los temas de sociogramas y el análisis de redes sociales, es esencial la lectura obligatoria de los recursos: *El análisis de las interacciones grupales: las aplicaciones SOCIOS* e *Introducción al Análisis de Redes Sociales*. De forma optativa se sugiere la lectura: *El sociograma*.

El docente dividirá al grupo en subgrupos al **azar** con un máximo de **cuatro integrantes**. En una *wiki* se analizarán cuáles son las **conexiones digitales** existentes entre los miembros del subgrupo, sea en *Facebook*, *Twitter* u otra red social de sugestión común, respondiendo a la pregunta: **¿qué conexiones poseen entre sí** que consideren relevantes en su vida social en la *Internet*? Con la información recopilada y discutida los alumnos construirán un **grafo social** donde cada uno constituya un nodo. Se podrán **añadir nodos** que representen otras personas que no estén en el subgrupo, páginas *Web*, intereses, juegos o cualquier otro aspecto de relevancia. El grafo deberá reflejar **todas las conexiones posibles** entre los **participantes** por lo que se espera posea una **cantidad** de nodos y aristas relativamente **significativa**.

Utilizando el software *Wolfram Mathematica* y la instrucción **Grafo** del paquete **VilCretas**, los estudiantes construirán un **grafo social 3D** con la información recolectada en el paso anterior. **Clasificarán** el grafo construido en alguno de los tipos vistos en clase: completo, bipartito, mariposa, rueda, estrella, círculo, camino, regular, red, entre otros. Usando los grados o valencias **determinarán** la cantidad de aristas que lo caracterizan. Con el grafo diseñado, se **responderán** las siguientes preguntas **justificando** teóricamente y a través del uso del paquete **VilCretas** como recurso de apoyo: ¿posee circuitos de *Euler*? y ¿posee circuitos de *Hamilton*?

Los estudiantes **reconstruirán** el grafo creado asignando **pesos** a cada una de las aristas en función del **nivel de importancia** en las relaciones sociales entre los nodos. Dos nodos débiles pero vinculados con una arista, podrían tener un peso 1, por el contrario, dos nodos con una relación fuerte podrían tener asociado un valor de 100, las escalas **serán definidas por cada subgrupo**, explicando en detalle el criterio utilizado para ello. Posteriormente, se seleccionarán **dos nodos no adyacentes** en el grafo social, con el objetivo de **aplicar** sobre ellos el algoritmo de la longitud del camino más corto, para determinar una **ruta de longitud más larga** usando el paquete **VilCretas** y el software *Wolfram Mathematica* ¿Qué representa el peso máximo de la ruta? Se sugiere consultar el siguiente video explicativo, sobre el empleo de un valioso comando, llamado **AIDijkstra**:



Finalmente, los estudiantes **responderán** a la pregunta: ¿es posible resolver el problema del agente viajero (*PAV*) en el grafo creado?, si es así, se deberá **devolver** dicha solución y en caso contrario, **modificar** el grafo con la intención de poder dar respuesta al *PAV*, explicando el criterio de alteración en el grafo original. En cualquier caso, se **animará visualmente** la solución utilizando la instrucción **AnimarGrafo**. Para **más detalles** sobre el uso de este comando se puede consultar:



Se solicitará a cada subgrupo la **elaboración de un video** que explique la presentación de su grafo social tridimensional y el proceso realizado en cada una de las **etapas** ya descritas. El video será publicado en: *YouTube* o *Vimeo*, con una duración máxima de cinco minutos y mínima de cuatro. El video deberá tener una **buena calidad** visual y de audio e integrar un guion **llamativo** y **original**. El enlace a la producción audiovisual se entregará por medio de un **espacio de tarea** al finalizar la semana el día domingo, por parte del **coordinador** de grupo.

Todo el proceso requerido para llevar a **buen término** las actividades ya compartidas, deberá quedar **consignado** en su totalidad en la *wiki* de trabajo, mostrando los distintos avances, formas de resolución, dominio de los contenidos y construcción conjunta de tesis. Las clases presenciales servirán como insumo para la **aclaramiento de dudas** y **defensa** de las propuestas al responder a las consultas del docente respecto a los **avances**. Los alumnos tendrán la responsabilidad de **demostrar** un adecuado desempeño teórico, práctico y técnico. El día miércoles se realizará en esta aula virtual un **chat** de consultas cuya participación es de **naturaleza optativa**.

Finalmente, en el **foro** vinculado a la clase se investigará de manera individual distintas herramientas para el análisis de sociogramas y redes sociales, dando **amplitud** a los temas tratados. Se insta a los estudiantes a participar en el **foro** desde el **primer día** de la semana (lunes) con el objetivo de garantizar **interacciones** oportunas y apropiadas. Se dará una respuesta **puntual** a lo solicitado y se **revisará** minuciosamente la redacción de las ideas y la ortografía. **No es válido** incluir una participación mediante un archivo adjunto.

La evaluación del *Taller 7 de resolución de problemas* se sustenta en la rúbrica desarrollada dentro de la *Guía Didáctica* del presente curso. Los criterios son:

- **Creación crítica y reflexiva de las soluciones:** los estudiantes resuelven de manera creativa e inédita las soluciones a los problemas propuestos y, el diseño y desarrollo del video.
- **Uso de recursos y software especializado:** los alumnos utilizan recursos didácticos y software como mecanismos de construcción y consulta.
- **Complejidad del contenido:** el video elaborado y las formas de resolución en la *wiki* son completas y dan una respuesta creativa y puntual.
- **Organización y secuencia:** los estudiantes presentan de forma organizada el contenido en el video y la *wiki*, evidenciando una secuencia lógica y ordenada entre cada una de las partes.
- **Originalidad:** el video es original, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.
- **Uso del tiempo:** el avance del trabajo colaborativo en la *wiki* y el video se realizan en el tiempo dispuesto por el docente.
- **Calidad de las participaciones en la wiki:** las participaciones en la *wiki* demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.
- **Participación en el foro:** los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio de los temas y una intervención constructiva.
- **Planificación previa y trabajo en clase:** los estudiantes planifican previamente junto al profesor las soluciones a los problemas propuestos y, el diseño y desarrollo del video. El trabajo en clase resuelto por los alumnos tiene un buen desempeño.
- **Dominio en el uso del software Wolfram Mathematica:** los alumnos demuestran dominio en el empleo del software *Wolfram Mathematica* y el paquete *VilCretas*.



- **Wiki:**

Subgrupo 1-Wiki: Taller 7 de resolución de problemas

Analicen cuáles son las **conexiones digitales** existentes entre los miembros del subgrupo, sea en *Facebook*, *Twitter* u otra red social de sugestión común, **respondiendo** a la pregunta: ¿qué conexiones poseen entre sí que consideren relevantes en su vida social en la *Internet*? Con la información recopilada y discutida **construyan** un grafo social donde cada uno constituya un nodo. Se **podrán añadir** nodos que representen otras personas que no estén en el subgrupo, páginas *Web*, intereses, juegos o cualquier otro aspecto de relevancia. El grafo deberá reflejar **todas** las conexiones posibles entre los participantes por lo que se espera posea una cantidad de nodos y aristas relativamente **significativa**. Utilizando el software *Wolfram Mathematica* y la instrucción **Grafo** del paquete **VilCretas**, construyan un grafo social *3D* con la información recolectada. **Clasifiquen** el grafo construido en alguno de los tipos vistos en clase: completo, bipartito, mariposa, rueda, estrella, círculo, camino, regular, red, entre otros. Usando los grados o valencias **determinen** la cantidad de aristas que lo caracterizan. Con el grafo diseñado, respondan las siguientes preguntas **justificando teóricamente** y a través del uso del paquete **VilCretas** como recurso de apoyo: ¿posee circuitos de *Euler*? y ¿posee circuitos de *Hamilton*? Ahora se **reconstruirá** el grafo creado asignando pesos a cada una de las aristas en función del **nivel de importancia** en las relaciones sociales entre los nodos. Dos nodos débiles pero vinculados con una arista, podrían tener un peso **1**, por el contrario, dos nodos con una relación fuerte podrían tener asociado un valor de **100**, las escalas serán **definidas** por cada subgrupo, **explicando** en detalle el criterio utilizado para ello. Posteriormente, seleccionarán dos nodos no adyacentes en su grafo social, con el objetivo de aplicar sobre ellos el **algoritmo de la longitud del camino más corto**, para determinar una **ruta de longitud más larga** usando el paquete **VilCretas** y el software *Wolfram Mathematica* ¿Qué representa el peso máximo de la ruta? Finalmente, se responderá a la pregunta: ¿es posible resolver el problema del agente viajero (*PAV*) en el grafo creado?, si es así, deberán **devolver** dicha solución y en caso contrario, **modificarán** el grafo con la intención de poder dar respuesta al *PAV*, explicando el **criterio de alteración** en el grafo original. En cualquier caso, **animarán visualmente** la solución utilizando la instrucción **AnimarGrafo**. Cada subgrupo se abocará en la **elaboración** de un **video** que explique la presentación de su **grafo social** tridimensional y el **proceso** realizado en cada una de las etapas ya descritas. El video será **publicado** en: *YouTube* o *Vimeo*, con una **duración** máxima de cinco minutos y mínima de cuatro.

Ver Editar Comentarios Historia Mapa Ficheros Administración

Grupos separados: Todos los participantes
 [Versión imprimible](#)

Trabajo colaborativo

Sociogramas y redes sociales

Esta página esta destinada a realizar el trabajo colaborativo semanal, ¡adelante con sus contribuciones!

- **Espacio de entrega de tarea:**

Espacio de entrega: Video

En este espacio se entregará el enlace a la **producción audiovisual** diseñada y desarrollada. El video será publicado en: *YouTube* o *Vimeo*, con una duración máxima de cinco minutos y mínima de cuatro. Deberá tener una **buena calidad** visual y de audio e integrar un guion **llamativo** y **original**.

Sumario de calificaciones

Participantes	0
Enviados	0
Pendientes por calificar	0

Ver/Calificar todas las entregas

Calificación


▪ Foro:

Foro: Herramientas de análisis de sociogramas y redes sociales

Este foro asistirá en el **estudio** de distintas herramientas para el **análisis de sociogramas y redes sociales**.



Añadir un nuevo tema de discusión

Tema	Comenzado por	Réplicas	Último mensaje
Consigna del foro	 ENRIQUE VILCHEZ QUESADA 401650194	0	ENRIQUE VILCHEZ QUESADA 401650194 Tue, 1 de May de 2018, 08:17

Consigna del foro

Suscrito

Mostrar respuestas anidadas ▾

Mover este tema a... ▾

Mover

Fijar



Consigna del foro

de ENRIQUE VILCHEZ QUESADA 401650194 - Tuesday, 1 de May de 2018, 08:17

Este foro tiene como objetivo **estudiar** herramientas de análisis de sociogramas. Se iniciará indagando en la *Web* distintas **herramientas** disponibles para el **análisis** de sociogramas y redes sociales. Realicen pruebas, sean creativos, utilicen sus propias redes sociales para *i*experimentar! Como sugerencia se les comparte este interesante *blog*: <http://ars-uns.blogspot.com/2016/06/30-paquetes-de-software-de-ars.html>. Seleccionen al menos **tres herramientas** y publiquen en este espacio una **justificación** teóricamente fundamentada de sus elecciones y algunos *screen shots* como evidencia del trabajo exploratorio realizado al **utilizar** cada software. La tarea deberá ser concretada en un plazo de **una semana** comenzando el día lunes y finalizando el día domingo ¡Se valorará la **interacción** entre las participaciones y el **uso creativo** de las herramientas de análisis!



[Enlace permanente](#) | [Editar](#) | [Borrar](#) | [Responder](#)

DOCUMENTOS ELABORADOS

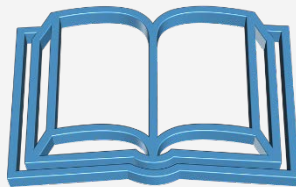
**UNIVERSIDAD NACIONAL DE COSTA RICA
ESCUELA DE INFORMÁTICA**



Curso:

***EIF-203 Estructuras Discretas para Informática bajo un
enfoque bimodal***

GUÍA DIDÁCTICA



Profesor:

Enrique Vílchez Quesada



1. Fundamentación de la materia

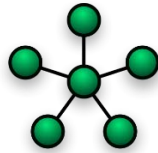
El curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal integra interesantes **definiciones y resultados fundamentales** de la **matemática finita** proporcionando un cimiento **teórico-conceptual** para abordar a futuro importantes áreas de la **informática** y las **ciencias computacionales**, como lo son: los sistemas operativos, las estructuras de datos, los lenguajes formales y los paradigmas de programación.

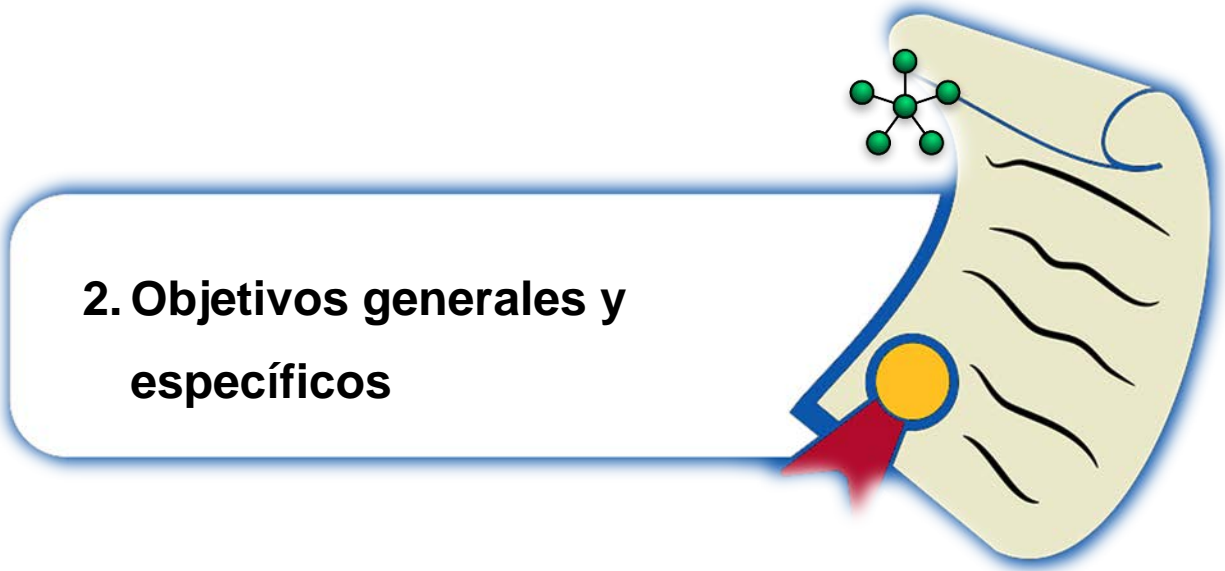
La materia aporta al estudiante las **estructuras discretas más importantes** para el desarrollo de soluciones computacionales eficientes, siendo esta área disciplinaria las **matemáticas por excelencia** utilizadas por los futuros profesionales en este campo. Las computadoras tratan con **objetos finitos** cuyo abordaje formal se hace posible a través de muchos de los **modelos no continuos y contables** suministrados por la **matemática discreta**.

Las experiencias de aprendizaje del curso se enfocan en la solución creativa ante problemas que abordan **situaciones vinculadas con la ingeniería en sistemas y ciencias de la computación**. Se pretende que el alumno **fomente destrezas y habilidades cognitivas** relacionadas con la inteligencia abstracta en la resolución de problemas, la argumentación de contenido matemático, la comunicación asertiva de

conclusiones de manera escrita, oral y audiovisual (uso adecuado de símbolos y representaciones), la capacidad en la modelización matemática de problemas simulados y la inventiva en el uso disciplinar de los algoritmos vistos en clase ante posturas abiertas y específicas.

El aprendizaje de las distintas teorías tendrá un **claro carácter significativo**, aplicando los métodos estudiados mediante una didáctica que provea los elementos necesarios hacia la generación de **inferencias, transformación y uso del conocimiento** en marcha, en consistencia con el desarrollo de **competencias tácitas** y el nivel curricular en el que se ubican los participantes.





2. Objetivos generales y específicos

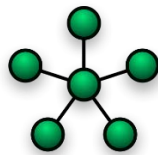
Objetivos generales:

- a. Conocer y entender las **teorías básicas de las estructuras discretas** para informática.
- b. Interpretar y aplicar las teorías básicas de las estructuras discretas para informática, en la **resolución de problemas** que enfrenten situaciones vinculadas con las **ciencias de la computación y la ingeniería en sistemas de información**.

Objetivos específicos:

- a. Conocer teorías básicas de la matemática discreta para informática, como lo son: aplicaciones de la **recursividad** en la programación, resolución de **relaciones de recurrencia, análisis de algoritmos, relaciones, grafos, árboles, autómatas de estado finito y, lenguajes y gramáticas** especiales.
- b. Comprender las **aplicaciones de la recursividad** dentro de la programación.
- c. Conocer distintos **métodos de resolución** de relaciones de recurrencia.

- d. Utilizar los principios de las notaciones asintóticas para el **análisis de la complejidad** de un algoritmo.
- e. Conocer la **teoría básica de las relaciones** y su vinculación con las funciones.
- f. Comprender los **aspectos esenciales** de la **teoría de grafos**: definición, representación, trayectorias, ciclos y algoritmos.
- g. Utilizar la **teoría de grafos** para **resolver problemas** orientados a las **ciencias de la computación**.
- h. Conocer los **aspectos principales** de la **teoría de árboles**: terminología, teoremas de caracterización, árboles de expansión, árboles binarios y sus aplicaciones como estructuras de datos.
- i. Utilizar los **árboles** y sus algoritmos para **resolver problemas** relacionados con la **informática**.
- j. Conocer la **teoría de autómatas de estado finito, lenguajes, gramáticas especiales** y sus relaciones.
- k. Emplear las teorías básicas de la matemática finita en la **resolución de problemas simulados** en los campos de las ciencias de la computación y la ingeniería en sistemas de información.
- l. Desarrollar **competencias digitales**, tales como: búsqueda, selección y análisis de fuentes, creación de nuevos nodos informativos, capacidades de autonomía en la toma de decisiones, síntesis de información, publicación e intercambio de nuevos resultados, ética, responsabilidad, uso de software especializado para la matemática discreta y habilidades de aprender a aprender, saber hacer y aprender en compañía de otros.





3. Contenidos

El curso se encuentra constituido por **ocho unidades temáticas principales**. Estas unidades se describen a continuación:

- **Recursividad**

- Definiciones recursivas.
- Aplicaciones de la recursividad a la programación.
- Comparación con la programación iterativa.

- **Relaciones de recurrencia**

- Definiciones básicas.
- Ejemplos de recurrencias y su solución.
- Método iterativo (sustitución hacia atrás o hacia adelante).
- Resolución de recursividades homogéneas lineales.
- Otros ejemplos de resolución de relaciones de recurrencia.

- **Análisis de algoritmos**

- Definición de algoritmo.
- Ejemplos de algoritmos.

- Criterios de análisis.
- Tiempo de corrida y recurrencias.
- Notaciones asintóticas: O , Ω y θ .
- Análisis gráfico de las notaciones asintóticas O , Ω y θ .
- Propiedades de las notaciones asintóticas O , Ω y θ .
- Aplicaciones de las notaciones O , Ω y θ al análisis de algoritmos.



▪ Relaciones

- Concepto de relación.
- Concepto de función a través de una relación.
- Representación digráfica de una relación.
- Ejemplos de tipos de relaciones.

▪ Grafos

- Concepto de grafo, digrafos y sus propiedades.
- Tipos de grafos.
- Caminos y circuitos: aplicaciones en informática.
- Circuitos de *Euler*, algoritmos, teoremas y aplicaciones.
- Circuitos de *Hamilton*, teoremas relacionados (*Dirac* y *Ore*) y aplicaciones sobre el problema del agente viajero.
- Matriz de adyacencia e incidencia y sus propiedades.
- Concepto de grafo ponderado.
- Algoritmo del camino más corto (*Dijkstra*).
- Problemas y aplicaciones clásicas de grafos.

▪ Árboles

- Concepto de árbol y sus propiedades.
- Árboles binarios y sus propiedades.
- Árboles generadores o de expansión y algoritmos buscar primero a lo ancho y a lo largo.

- Operaciones de los árboles binarios: búsqueda, inserción y recorridos preorden, interorden y postorden.
- Árboles con etiquetas y sus aplicaciones sobre la representación de expresiones algebraicas.
- Árboles de expansión mínima y algoritmos para su generación.
- Problemas y aplicaciones clásicas de árboles.

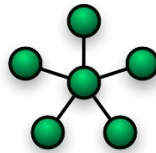
■ Máquinas de estado finito

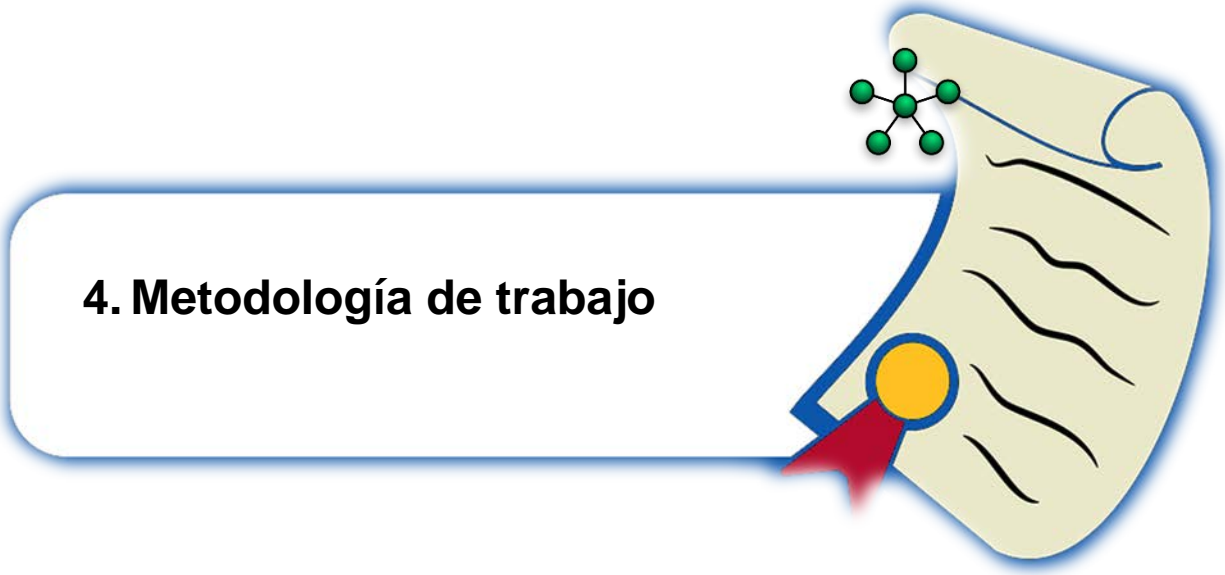
- Concepto y propiedades de una máquina de estado finito.
- Concepto y propiedades de un autómata de estado finito.
- Recorridos de hileras en un autómata de estado finito.
- Construcción de autómatas de estado finito.



■ Lenguajes y gramáticas

- Concepto y propiedades de los lenguajes y las gramáticas.
- Conjuntos de hileras y operaciones.
- Casos de lenguajes/expresiones regulares y lenguajes/gramáticas libres de contexto.
- Relaciones entre los lenguajes regulares y los autómatas finitos.





4. Metodología de trabajo

El logro de los objetivos del presente curso se sustentará en un abordaje de los contenidos basado en la siguiente **planificación pedagógica**:

- **Presentaciones magistrales** a cargo del profesor en las sesiones presenciales.
- Desarrollo de **ejercicios teórico-prácticos y aplicados** en el campo de la informática de forma individual y grupal.
- Utilización de **herramientas de software** para abarcar algunas de las áreas de contenido de la materia, tanto en sus aspectos teóricos como prácticos.
- Promoción de una **investigación constructiva** en los estudiantes, de tal manera que éstos brinden aportes al **aprendizaje propio** y de los **demás** y, construyan sus propias soluciones a algunos problemas planteados.
- Realización de diversas **actividades** para facilitar una **asimilación** más propicia y **profundización** de cada una de las unidades temáticas del curso. Dichas actividades se enmarcan en: **talleres** de resolución de problemas, una **exposición grupal** de un tema de interés, **comprobaciones** de lectura, un **proyecto programado**, un **proyecto de diseño y desarrollo** de una producción audiovisual, participación en **foros de discusión** y realización de **pruebas escritas**.

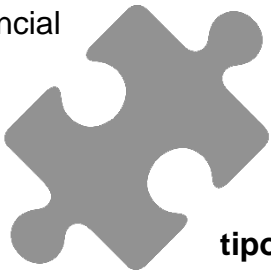
Las actividades se propondrán a la población estudiantil usando un **patrón de medida semanal**. El **inicio** de semana se efectuará los **lunes** y el **final** de semana los días **domingo**.

Los **talleres de resolución de problemas** brindan un espacio colaborativo a los alumnos para **emplear** distintos recursos didácticos y esquemas de **razonamiento** con la meta de proponer de manera grupal **soluciones creativas, críticas y reflexivas** a problemas vinculados con la matemática finita. Los talleres tendrán una duración de **una semana** combinando labores en la clase presencial y el aula virtual institucional. Cada taller tendrá asociado la **activación de una clase** en la plataforma *Moodle* donde se incorporarán las instrucciones específicas a seguir. Se abrirá un espacio de **tarea** para la entrega grupal correspondiente al **finalizar la semana**. El **proceso completo** de resolución de las situaciones problema será consignado en el aula virtual, dejando un registro de las labores colaborativas efectuadas gestionado a través de una **wiki** facilitada por el docente al **inicio de la actividad**. En la clase presencial se brindará un soporte de seguimiento a cada uno de los subgrupos de trabajo. Virtualmente los mecanismos de consulta serán: la **mensajería interna**, el **chat** y la **videoconferencia** de la plataforma *Moodle*, los cuales, para efectos de esta tarea, son de **participación voluntaria**.



El profesor asignará por subgrupos al **inicio del curso**, un **tema** de la materia con la intención de ser **expuesto** en un formato **presencial**. Cada subgrupo analizará cómo abordar el contenido bajo un **perfil distinto** a lo planificado por el docente en sus lecciones magistrales, con la finalidad de realizar un aporte **inédito** a los objetivos de la asignatura. Los alumnos buscarán **fuentes de información** de **autores reconocidos** a nivel nacional e internacional. Además, se indagará cuáles **tecnologías digitales o software especializado** pueden proporcionar un apoyo en las tareas comprometidas hacia la aplicación del contenido en la **resolución de problemas** inmersos en el campo de la **informática**. Como productos se espera un **documento escrito** y una **presentación**, a **entregarse** a través del aula virtual institucional mediante un **espacio**

de tarea, al **finalizar** la **semana previa** a la exposición presencial de acuerdo con el **cronograma** del curso. El proceso de **planificación**, **diseño** y **elaboración** se registrará de dos maneras distintas: por intermedio de la **construcción de un glosario de conceptos base** (en el aula virtual, abierto por el docente) y el empleo de una **wiki** donde se dejará en evidencia el **trabajo colaborativo** implementado. El glosario consistirá en un espacio virtual de generación de definiciones vinculadas con el tema a exponer, que servirá de guía conceptual para los demás compañeros del grupo. La **coevaluación** se constituirá en un elemento esencial de valoración conjunta, al **finalizar la semana** durante la cual cada subgrupo realizará su presentación. Para ello el docente abrirá un **espacio de taller** en el aula virtual institucional. En un plazo de **quince días previos** a la presentación presencial, los alumnos ejecutarán todas las etapas relacionadas con esta tarea bajo la guía y supervisión del profesor en las clases presenciales y la plataforma *Moodle*. Las consultas virtuales podrán ser planteadas de **forma optativa**, a través de: la **mensajería interna**, el **chat** y la **videoconferencia**.

En el aula virtual institucional se designará a los estudiantes **distintas lecturas** que deberán realizar de forma individual antes de la lección presencial correspondiente, durante la semana indicada dentro del cronograma de la materia. Con ello el docente partirá de un **conocimiento previo** que servirá de base en el desarrollo de sus lecciones magistrales. Al finalizar la semana se aplicará una **comprobación**  **tipo cuestionario** habilitada en el **espacio virtual**, con el objetivo de garantizar un compromiso de lectura **consciente** y **responsable** por parte de los alumnos.

El **proyecto programado** implementado en el **lenguaje Wolfram** se entregará de manera grupal al profesor al **finalizar el curso** (ver el cronograma). El docente en el aula virtual **mostrará ejemplos** de algoritmos que podrían ser elegidos por los distintos subgrupos de trabajo, en lenguajes de programación distintos al lenguaje *Wolfram*. El proyecto se **iniciará en la semana 12** (un mes antes de su entrega) mediante una **clase activada** por el profesor en el aula virtual. Los estudiantes en grupo completarán una secuencia de **planificación** fundamentada en: selección del algoritmo a programar y

justificación, diseño y desarrollo, depuración de errores y pruebas de ejecución. El programa será **defendido** de manera **presencial** explicando posibles **aplicaciones reales** en el área disciplinar de interés. Los productos finales serán **subidos** a un espacio de **entrega de tarea** en el aula virtual, creado por el docente con este objetivo. Dichos productos versarán en la **implementación del algoritmo** en el lenguaje *Wolfram* y un **archivo de presentación**, que serán subidos por uno de los miembros del subgrupo **una semana antes** de la exposición presencial del proyecto. El avance del trabajo colaborativo quedará suscrito mediante el uso de una **wiki** en el aula virtual y el docente se comprometerá en este espacio a dar un seguimiento adecuado brindando recomendaciones. Las dudas emergidas se evacuarán por medio de: la **mensajería interna**, el **chat** y la **videoconferencia**. El profesor abrirá distintos espacios de comunicación sincrónicos (*chat* y *videoconferencia*) de **participación obligatoria**. La *wiki* se empleará, además, como un espacio de discusión sobre las **fortalezas** y **debilidades** del algoritmo escogido en la resolución de problemas aplicados en la informática.

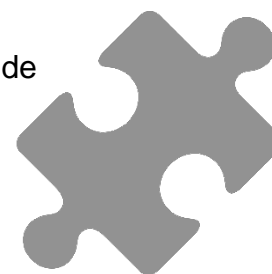
El proyecto de **diseño y desarrollo de una producción audiovisual** proporciona una oportunidad de selección de un **tema libre** de la materia para mostrar en un **video creativamente** y de manera **innovadora**, el contenido elegido analizando de forma **crítica** su importancia **instrumental** y **aplicada** en el campo de la informática. Se estimulará el uso de herramientas **Web 2.0** recomendadas por el docente en el aula virtual institucional. El video será realizado **grupalmente** a través de una **wiki** mostrando avances y estrategias de resolución. La tarea **iniciará** en la **semana 11** con una **duración de un mes**. En la **wiki**, también, los estudiantes expondrán **aplicaciones e importancia científica** del tema elegido en el campo de la ingeniería en sistemas de información y las ciencias computacionales. Al **finalizar**, en la **semana 16**, el video será publicado en una **red social** a escoger: **YouTube** o **Vimeo** y la dirección *URL* se **socializará** en la **wiki** y se compartirá con el profesor en un **espacio de entrega de tarea** dentro del aula virtual. Los recursos **optativos** para el planteamiento de consultas



fuera de las clases presenciales serán: la **mensajería interna**, el **chat** y la **videoconferencia**.

Los **foros de discusión** servirán como espacios promotores de una mayor **profundización** de las áreas temáticas del curso, serán valorados considerando: **cantidad y puntualidad** de las participaciones, **calidad** de los *posts* y colaboración en el favorecimiento de un **aprendizaje en red**. Se realizarán **tres foros** de discusión durante el curso, abarcando cada uno, las **tres partes principales** en las que se subdivide la materia, a saber:

- **Parte I:** recursividad, relaciones de recurrencia y análisis de algoritmos.
- **Parte II:** relaciones, teoría de grafos y teoría de árboles.
- **Parte III:** máquinas de estado finito, autómatas, lenguajes y gramáticas.



Cada foro tendrá una duración de **una semana** tal y como se especifica en el cronograma.

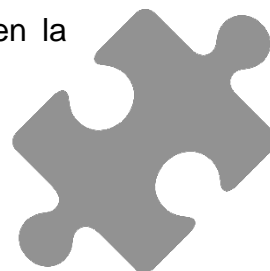
La **redacción y ortografía** en todas las actividades de aprendizaje expuestas se considera un **aspecto medular** en los entregables, por lo que los alumnos deberán tener un **control riguroso y estricto** sobre la forma en cómo expresan sus ideas, procesos de análisis y debates en concordancia directa con las **reglas gramaticales** del idioma español.

En todas las actividades descritas se visualiza al estudiante como una persona **altamente participativa y comprometida** con su propio aprendizaje, **responsable** y **honesto** al desarrollar cada una de las tareas implicadas en la materia. Se requiere que el alumno dedique al menos seis horas de estudio extra clase y asista a las horas de consulta del docente.

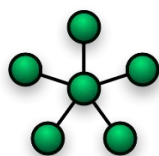
Se espera que el profesor sea un **guía** en el proceso de aprendizaje y que promueva el **carácter investigativo** del estudiante. En este sentido, la utilización de diferentes

recursos didácticos para desarrollar sus lecciones presenciales y virtuales se estima como un elemento fundamental hacia la búsqueda de una estructuración sistemática de **diferentes tipos de ambientes de aprendizaje**, tanto de carácter **algoritmo**, como de carácter **heurístico** (por descubrimiento). Por medio del uso del **aula virtual institucional**, **software** u otras **estrategias didácticas**, el profesor complementará sus lecciones de corte tradicional, con la **visualización** conceptual, **recorridos** de algoritmos, **presentación** del contenido de una forma más **dinámica** y, sobre todo, la **solución a problemas** vinculados con la **ingeniería en sistemas de información y ciencias computacionales**. En general, los mecanismos de comunicación no presencial con el docente y los otros participantes se constituirán en: la **mensajería interna**, el **chat** y la **videoconferencia** de la plataforma *Moodle*.

El profesor promoverá las experiencias de aprendizaje de la materia implicando el **desarrollo de habilidades inventivas, creativas y analíticas**, en la ejercitación práctica ante problemas que el alumno delimitará, razonará y resolverá en condiciones mediadas por el docente o el mismo estudiante. Por otra parte, mediante la implementación de las diversas actividades de aprendizaje ya señaladas, el alumno mejorará sus **habilidades metacognitivas**, formas de **comprensión** de un problema y tareas que le permitan **aprender a aprender**.



El **material del curso** estará disponible para los estudiantes en el **aula virtual institucional**, cuya dirección *URL* es la siguiente: <http://www.aulavirtualid.una.ac.cr/login/index.php>. Además, en: <http://www.escinf.una.ac.cr/discretas> se habilitarán otros materiales **complementarios** disponibles.





5. Evaluación de los aprendizajes

El trabajo será **evaluado** por medio de **tres pruebas parciales** y distintas actividades de aprendizaje, a saber: **ocho talleres de resolución de problemas**, **una exposición grupal** de un tema del curso, **ocho comprobaciones** de lectura a realizar en el aula virtual, **un proyecto programado**, **un proyecto de diseño y desarrollo** de una **producción audiovisual** con tema libre y participación en **tres foros** de discusión.

Las **pruebas parciales** serán aplicadas en el **tiempo de clase presencial** durante la semana indicada en el cronograma. Las **comprobaciones de lectura** se suministrarán bajo una **modalidad de cuestionario** dentro del aula virtual institucional según el calendario de la materia.

Los **talleres de resolución de problemas** serán evaluados a través de la siguiente **rúbrica**:

Criterios de Evaluación	10 Excelente	8 Bueno	6 Regular	4 Deficiente	2 Pobre
1. Creación crítica y reflexiva de las soluciones					

<p>Los estudiantes resuelven de manera creativa e inédita las situaciones propuestas.</p>					
<p>2. Uso de recursos y software especializado</p> <p>Los alumnos utilizan recursos didácticos y software como mecanismos de solución.</p>					
<p>3. Completitud del contenido</p> <p>Las soluciones propuestas son completas y dan una respuesta creativa y puntual a lo planteado.</p>					
<p>4. Organización y secuencia</p> <p>Los estudiantes presentan de forma organizada el contenido, evidenciando una secuencia lógica y ordenada entre cada una de las partes.</p>					
<p>5. Originalidad</p> <p>Las soluciones desarrolladas son originales, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.</p>					
<p>6. Uso del tiempo</p> <p>Los entregables se realizan en el tiempo dispuesto por el docente.</p>					



<p>7. Calidad de las participaciones en la <i>wiki</i></p> <p>Las participaciones en la <i>wiki</i> demuestran un trabajo colaborativo apropiado con aportes abocados hacia la profundización de la materia.</p>					
<p>8. Participación en el foro</p> <p>Los alumnos participan en el foro vinculado a la clase, demostrando un buen dominio de los temas y una intervención constructiva.</p>					
<p>9. Planificación previa y trabajo en clase</p> <p>Los estudiantes planifican previamente junto al profesor sus propuestas de solución. El trabajo en clase resuelto por los alumnos tiene un buen desempeño.</p>					
<p>10. Dominio en el uso del software <i>Wolfram Mathematica</i></p> <p>Los alumnos demuestran dominio en el empleo del software <i>Wolfram Mathematica</i> para la resolución de los problemas dispuestos.</p>					



La **exposición grupal** de un tema de interés de la materia será evaluada a través del uso de los siguientes **criterios**:

Criterios de Evaluación	10 Excelente	8 Bueno	6 Regular	4 Deficiente	2 Pobre
1. Calidad de la presentación El contenido de la presentación brinda un aporte distinto e inédito. Los estudiantes mantienen la atención de los espectadores.					
2. Dominio del contenido y seguridad de los planteamientos Los alumnos demuestran dominio del contenido curricular (evitan limitarse a leer) y proyectan seguridad en cada uno de los planteamientos, evidenciado por su exposición y la respuesta a las preguntas de los asistentes.					
3. Organización y secuencia Los estudiantes presentan de forma organizada el contenido. Se evidencia una secuencia lógica y ordenada entre cada una de las partes.					
4. Calidad del documento escrito y archivo con la presentación					



<p>La calidad del documento escrito final y archivo con la presentación es adecuada y las fuentes de información utilizadas corresponden a autores reconocidos a nivel nacional e internacional.</p>					
<p>5. Uso del tiempo Los alumnos utilizan adecuadamente el tiempo disponible para su presentación y hacen entrega de los productos puntualmente.</p>					
<p>6. Uso de recursos visuales, tecnológicos y software especializado Los estudiantes hacen uso adecuado de recursos visuales y tecnológicos enriqueciendo su presentación. Además, muestran el empleo de software especializado para la aplicación del contenido en la resolución de problemas.</p>					
<p>7. Calidad de las participaciones en la <i>wiki</i> Las participaciones en la <i>wiki</i> demuestran un trabajo colaborativo apropiado con</p>					



aportes abocados hacia la profundización de la materia.					
8. Calidad de las participaciones en el glosario de conceptos Las participaciones en el glosario de definiciones demuestran un trabajo colaborativo apropiado y una profundización de los contenidos.					
9. Planificación previa Los alumnos planifican previamente junto al profesor su exposición y todas las actividades llevadas a cabo.					
10. Resultados de la coevaluación Los resultados de la coevaluación demuestran un aporte inédito de la exposición a los objetivos del curso.					



El **proyecto programado** se evaluará tomando en consideración la **rúbrica** expuesta a continuación:

Criterios de Evaluación	10 Excelente	8 Bueno	6 Regular	4 Deficiente	2 Pobre
1. Creación crítica y reflexiva del código de programación Los estudiantes resuelven de manera creativa e inédita la					

implementación en el lenguaje <i>Wolfram</i> del algoritmo elegido.					
2. Uso del lenguaje <i>Wolfram</i> Los alumnos demuestran dominio y un buen nivel de profundización en el empleo del lenguaje de programación <i>Wolfram</i> .					
3. Completitud La implementación del algoritmo es completa y da una respuesta validada, creativa y puntual al problema que se aborda.					
4. Organización y secuencia Los estudiantes presentan el código de programación de forma organizada, indentado y con comentarios evidenciando una secuencia lógica y ordenada entre cada una de las partes.					
5. Originalidad Las soluciones desarrolladas son originales, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.					
6. Uso del tiempo					



Los entregables se realizan en el tiempo dispuesto por el docente.					
7. Calidad de las participaciones en la <i>wiki</i> Las participaciones en la <i>wiki</i> demuestran un trabajo colaborativo apropiado, completando las etapas de: selección del algoritmo y justificación, diseño y desarrollo, depuración de errores, pruebas de ejecución e identificación de fortalezas y debilidades del algoritmo en la resolución de problemas.					
8. Participación en los espacios de aclaración de dudas Los alumnos participan en los <i>chat</i> y las <i>videoconferencias</i> de evacuación de dudas.					
9. Planificación previa Los estudiantes planifican previamente junto al profesor sus propuestas de solución.					
10. Calidad de la presentación presencial El contenido de la presentación presencial de la implementación del algoritmo proporciona un aporte inédito					



al explicar sus posibles aplicaciones, fortalezas y debilidades en la resolución de problemas.					
--	--	--	--	--	--



El proyecto de diseño y desarrollo de una producción audiovisual será valorado bajo los siguientes criterios:

Criterios de Evaluación	10	8	6	4	2
	Excelente	Bueno	Regular	Deficiente	Pobre
1. Calidad de la presentación en el video Los estudiantes mantienen la atención de los espectadores. El video no resulta aburrido.					
2. Dominio y completitud del contenido Los estudiantes demuestran dominio y seguridad en el contenido curricular. El video es completo, crítico, innovador y creativo en cuanto a la información que ofrece.					
3. Organización y secuencia Los estudiantes presentan de forma organizada el contenido del video, evidenciando una secuencia lógica y ordenada entre cada una de las partes.					

<p>4. Originalidad</p> <p>El video es original en cuanto a su contenido teórico y audiovisual, empleando recursos propios, o bien, respetando los derechos de autor con relación al material reutilizado.</p>					
<p>5. Uso del tiempo</p> <p>El video presenta una duración máxima de 5 minutos y mínima de 4 minutos.</p>					
<p>6. Calidad de los recursos visuales</p> <p>Los recursos visuales empleados presentan claridad, buena definición, suficiente luz y una edición apropiada.</p>					
<p>7. Calidad del audio</p> <p>El audio integrado es claro, con un volumen suficiente y adecuado, no posee cortes bruscos ni interrupciones auditivas.</p>					
<p>8. Planificación previa</p> <p>Los estudiantes planifican previamente junto al profesor el diseño y</p>					



desarrollo del video, elaborando una propuesta fundamentada.					
9. Repaso y profundización El video fortalece los contenidos teóricos, prácticos y aplicados del curso, promoviendo un espacio de repaso y profundización de la materia.					
10. Calidad de las participaciones en la <i>wiki</i> y publicación Las participaciones en la <i>wiki</i> demuestran un trabajo colaborativo apropiado en todas las etapas del diseño y desarrollo del material audiovisual. El video es publicado en <i>YouTube</i> o <i>Vimeo</i> y socializada en la <i>wiki</i> su dirección <i>URL</i> .					



Además, los **tres foros de discusión** en el aula virtual se **evaluarán** recurriendo a los siguientes **rubros** (todos con la misma ponderación):

- **Cantidad y puntualidad de las participaciones:** mínimo dos por alumno, la primera el lunes o martes y la última el jueves o viernes. Los miércoles el profesor moderará las participaciones y el sábado o domingo realizará un cierre utilizando las ideas principales de la discusión.

- **Calidad de las participaciones:** en términos de consistencia y profundidad a las temáticas del curso. Se recomienda investigar en la Red acerca de los contenidos tratados u otros afines.
- **Trabajo colaborativo:** las intervenciones no deben ser aisladas, promoviendo así, la discusión respetuosa y crítica de todos hacia una construcción colectiva de significados.

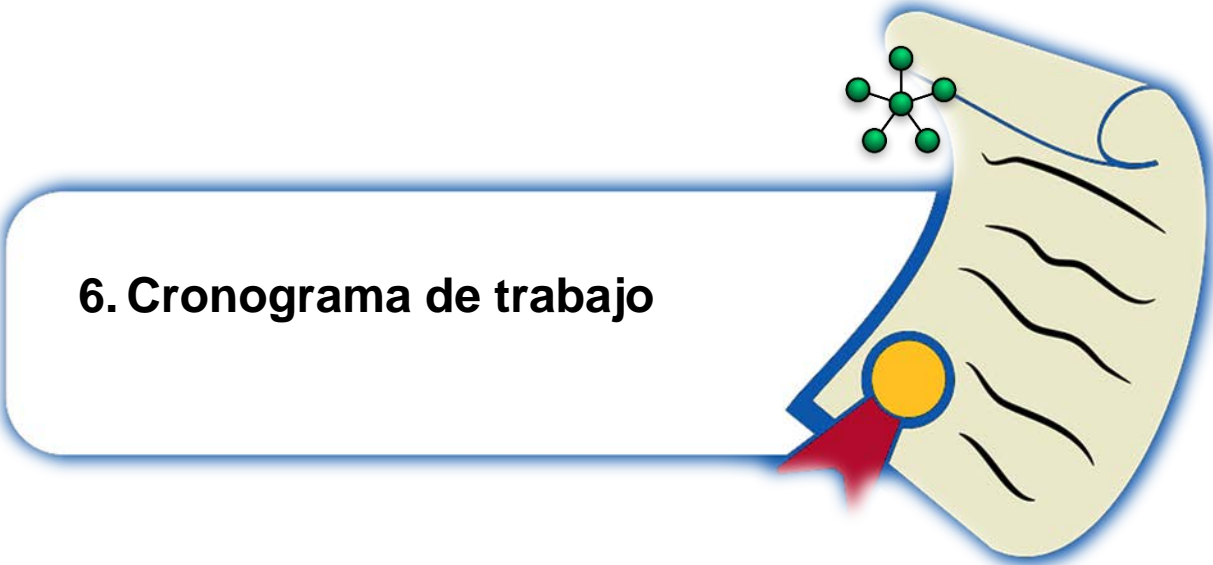
El **valor porcentual** de las experiencias de evaluación descritas se detalla a continuación:

Actividad de aprendizaje	Porcentaje de la nota final
Tres pruebas parciales (con el mismo valor porcentual)	3
Ocho talleres de resolución de problemas (con el mismo valor porcentual)	1.5
Una exposición grupal	1
Ocho comprobaciones de lectura (con el mismo valor porcentual)	1
Proyecto programado	1.5
Proyecto de diseño y desarrollo de una producción audiovisual	1
Participación en tres foros de discusión (con el mismo valor porcentual)	1



La **calificación mínima de aprobación** es un **7**. Si el estudiante **no aprueba** el curso, pero obtiene una **nota superior o igual a 6**, podrá presentar un **examen extraordinario** previo pago de los **derechos** correspondientes en el **Departamento de Financiero** de la universidad. Para aquellos alumnos que deban presentar el examen extraordinario, en su fecha de aplicación, deberán traer cancelado el **recibo del derecho respectivo**, sin este recibo el estudiante **no podrá** realizar la prueba.





6. Cronograma de trabajo

A continuación, se presenta el **cronograma cronológico de temas y actividades** a realizar durante el ciclo lectivo. Ya se encuentra **habilitado** un espacio común para dudas a través del **chat** y la **videoconferencia** en el aula virtual institucional. Además, los espacios para **entrega de tareas**, **wikis** de trabajo colaborativo, **glosario de definiciones** y **coevaluación** requeridas en las actividades de: talleres de resolución de problemas, exposición grupal, proyecto programado y realización de un video, están **disponibles en el aula virtual** para su uso, en el momento que los alumnos lo consideren oportuno en **mutuo acuerdo** con el profesor.

Tema a desarrollar	Actividades a aprendizaje	Duración semanal
Recursividad	<ul style="list-style-type: none"> ▪ Asignación de los temas por subgrupo para la exposición grupal: al iniciar la semana 1 ▪ Comprobación de lectura 1: al finalizar la semana 1 ▪ Inicio del trabajo colaborativo de la exposición grupal 1: semana 1 ▪ Taller 1 de resolución de problemas: semana 2 	2 (semanas 1 y 2)

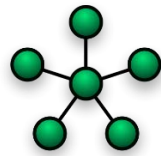


Relaciones de recurrencia	<ul style="list-style-type: none">▪ Exposición grupal 1: semana 3▪ Comprobación de lectura 4: al finalizar la semana 3▪ Inicio del trabajo colaborativo de la exposición grupal 2: semana 3▪ Taller 2 de resolución de problemas: semana 4▪ Inicio del trabajo colaborativo de la exposición grupal 3: semana 4	2 (semanas 3 y 4)
Análisis de algoritmos	<ul style="list-style-type: none">▪ Exposición grupal 2: semana 5▪ Comprobación de lectura 3: al finalizar la semana 5▪ Taller 3 de resolución de problemas: semana 6▪ Exposición grupal 3: semana 6▪ Inicio del trabajo colaborativo de la exposición grupal 4: semana 6	2 (semanas 5 y 6)
Relaciones	<ul style="list-style-type: none">▪ Examen parcial 1: semana 7▪ Comprobación de lectura 4: al finalizar la semana 7▪ Foro de discusión 1 (parte I del curso): semana 7▪ Taller 4 de resolución de problemas: semana 8▪ Exposición grupal 4: semana 8	2 (semanas 7 y 8)
Teoría de grafos	<ul style="list-style-type: none">▪ Taller 5 de resolución de problemas: semana 9▪ Inicio del trabajo colaborativo de la exposición grupal 5: semana 9▪ Taller 6 de resolución de problemas: semana 10	3 (semanas 9, 10 y 11)

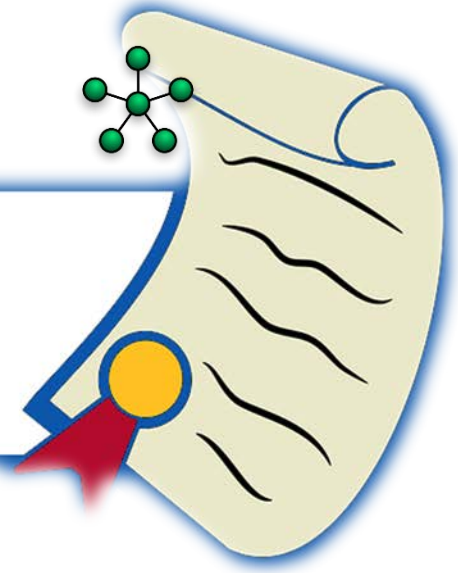


	<ul style="list-style-type: none">▪ Comprobación de lectura 5: al finalizar la semana 10▪ Taller 7 de resolución de problemas: semana 11▪ Exposición grupal 5: semana 11▪ Inicio del trabajo colaborativo de la exposición grupal 6: semana 11▪ Inicio del trabajo colaborativo del proyecto de diseño y desarrollo de una producción audiovisual: semana 11	
Teoría de árboles	<ul style="list-style-type: none">▪ Comprobación de lectura 6: al finalizar la semana 12▪ Inicio del trabajo colaborativo del proyecto programado: semana 12▪ Taller 8 de resolución de problemas: semana 13▪ Exposición grupal 6: semana 13▪ Inicio del trabajo colaborativo de la exposición grupal 7: semana 13	2 (semanas 12 y 13)
Máquinas y autómatas de estado finito	<ul style="list-style-type: none">▪ Examen parcial 2: semana 14▪ Comprobación de lectura 7: al finalizar la semana 14▪ Foro de discusión 2 (parte II de la materia): semana 14▪ Inicio del trabajo colaborativo de la exposición grupal 8: semana 14▪ Exposición grupal 7: semana 15	2 (semanas 14 y 15)
Lenguajes y gramáticas	<ul style="list-style-type: none">▪ Exposición grupal 8: semana 16▪ Comprobación de lectura 8: al finalizar la semana 16▪ Entrega del video: al finalizar la semana 16	1 (semana 16)

Cierre del curso	<ul style="list-style-type: none"> ▪ Examen parcial 3: semana 17 ▪ Presentación de los proyectos programados: semana 17 ▪ Foro de discusión 3 (parte III del curso): semana 17 ▪ Entrega de promedios: semana 18 ▪ Estudio independiente para el examen extraordinario: semana 19 ▪ Examen extraordinario: semana 20 	<p style="text-align: center;">4 (semanas 17, 18, 19 y 20)</p>
-------------------------	--	--



7. Presentación del tutor



Estimados y estimadas estudiantes: mi nombre es **Enrique Vílchez Quesada** soy docente e investigador titular a **tiempo completo** de la **Escuela de Informática** de la **Universidad Nacional de Costa Rica (UNA)**, con categoría de profesor catedrático. Cuento con una *Licenciatura en la Enseñanza de la Matemática* y una *Maestría en Tecnología e Informática Educativa*, ambos grados obtenidos con distintas menciones honoríficas en la **UNA**.



He laborado, además, como profesor a tiempo parcial en la **Escuela de Matemática** del **Instituto Tecnológico de Costa Rica (ITCR)** y la **Universidad de Costa Rica (UCR)**. He colaborado como **coordinador** y **participante** de distintos **proyectos de investigación** en la **Universidad Nacional de Costa Rica**, relacionados con el uso sistemático de las **tecnologías de información y comunicación (TIC)** en el ámbito **educativo** a nivel superior.

Cuento con una amplia experiencia docente y profesional desempeñándome en **distintos campos** del quehacer académico y administrativo, entre los que se destaca el cargo de **subdirector en la Escuela de Informática de la UNA** y una labor docente en **prestigiosas instituciones universitarias** en Costa Rica. He publicado más de

cuarenta artículos especializados en el campo de la **educación matemática** y **enseñanza de la informática** y participado como ponente y profesor invitado en más de **cincuenta congresos a nivel nacional e internacional**, en países tales como: Brasil, México, Uruguay, República Dominicana, Cuba, USA, Venezuela, Argentina, Colombia, Panamá, España y Costa Rica.

Mis principales **aportes académicos** y **docentes** abarcan diversidad de áreas en la **matemática** y sus aplicaciones, **tecnologías de información y comunicación**, aprendizaje **virtual** y fundamentos de **programación**. Mis más recientes obras se han enfocado en divulgar experiencias y metodologías innovadoras apoyadas en el **uso de TIC**, particularmente los libros: ***Álgebra lineal apoyada con Mathematica*** publicado por la editorial ***Tecnológica de Costa Rica*** y el texto ***Estructuras Discretas con Mathematica*** publicado por la editorial mexicana ***Alfaomega***.

Con **mucha emoción** quiero darles la **bienvenida al curso**, en este espacio ya pueden encontrar todo el **contenido** que desarrollaremos durante el semestre. Los invito a realizar una **revisión del material y vínculos** que he compartido, recomiendo que con tiempo bajen todos los archivos y vayan visualizando la información con **detenimiento**. Espero que podamos hacer un **excelente equipo de trabajo** y estoy a sus **órdenes** para lo que **requieran** a lo largo del presente ciclo lectivo.

Prof.
Enrique Vilchez



Saludos cordiales,

Prof. Enrique Vilchez Quesada



**UNIVERSIDAD NACIONAL DE COSTA RICA
ESCUELA DE INFORMÁTICA**



Curso:

***EIF-203 Estructuras Discretas para Informática bajo un
enfoque bimodal***

UNIDAD DIDÁCTICA:

Teoría de Grafos



Profesor:

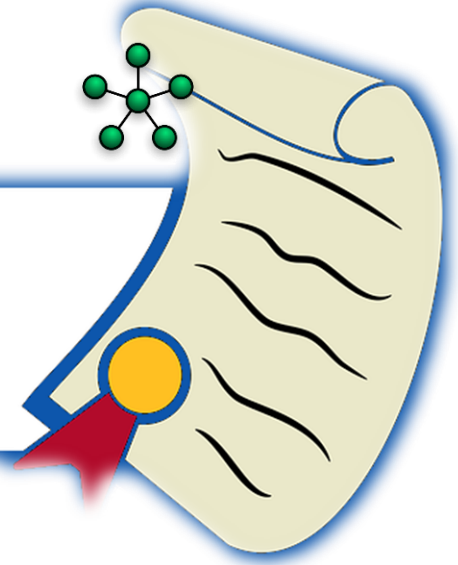
Enrique Vílchez Quesada

Contenidos

Introducción	3
Conceptos generales	6
Tipos de grafos	28
Representaciones para un grafo	40
Circuitos en un grafo	54
Algoritmo del camino más corto	72
Ejercicios	79
Bibliografía	82

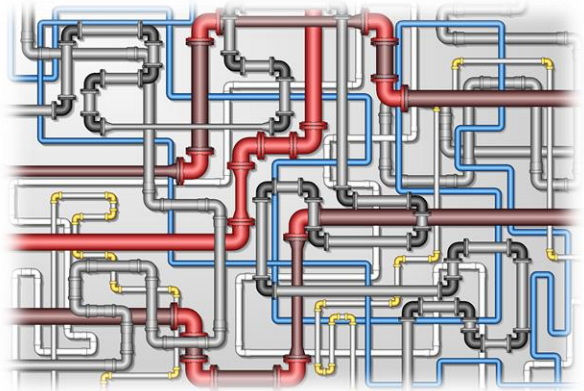


Introducción



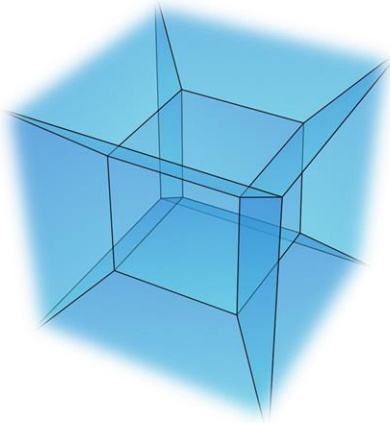
La teoría de grafos propone un conjunto de definiciones y propiedades que presentan diversas aplicaciones en la informática. Su tarea principal, reside en **simplificar muchos tipos de problemas** que, al ser representados por medio de un grafo, se resuelven con mayor facilidad. Además, la naturaleza misma de ciertas situaciones deriva en la formulación de un grafo, tal es el caso del **análisis de redes de transporte o de flujo**.

Otro ejemplo se circunscribe cuando se piensa en la manera en cómo se encuentran interconectadas una serie de computadoras en una **intranet**, al tomar una curva que las une, tal figura, genera gráficamente un grafo. La **estructura de las tuberías** en un sistema de acueductos constituye un ejemplo adicional en esta dirección.



También, en programación paralela, un conocido modelo denominado **hipercubo** (on -cubo), construye un grafo. El hipercubo es una forma de simulación de un procesador que cuenta con varios núcleos. La idea principal del hipercubo es representar las relaciones de los distintos núcleos de un mismo procesador, cuando se utiliza la programación paralela. Recibe este nombre, pues en ella, el programador optimiza el uso del procesador corriendo sus núcleos por separado y en paralelo, para mejorar los

tiempos de ejecución en la respuesta computacional. El grafo del n -cubo se forma al tomar 2^n puntos, llamados nodos o vértices, que representan cada uno, un núcleo del procesador. El 1-cubo posee dos nodos y una arista que los une, cada vértice se etiqueta con un número binario de un *bit*. El 2-cubo se construye tomando cuatro nodos (se asumen 4 núcleos), los dos ya existentes y dos puntos más, etiquetados con binarios de longitud dos. Como regla general, se conectarán en el n -cubo, aquellos vértices que



difieren en un *bit* con relación al $(n-1)$ -cubo. El 3-cubo se genera al tomar 2^3 núcleos, gráficamente esto forma un cubo en el espacio y cada vértice se etiqueta con un número binario de longitud tres. El 4-cubo se forma al tomar 2^4 nodos. Su construcción se resumiría intuitivamente como conectar por vértices dos cubos del espacio tridimensional, añadiendo las aristas necesarias y etiquetas de números binarios de longitud 4. El *software*

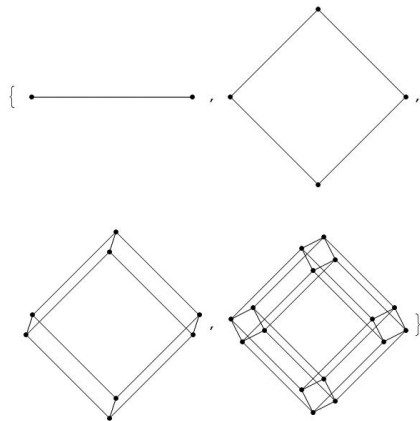
Mathematica en el que se encuentra basado el presente curso, posee el comando *Hypercube* que muestra el hipercubo de dimensión n . El siguiente ciclo, grafica los n -cubos con $n \in \{1, 2, 3, 4\}$:

`In[] :=`

`<<Combinatorica``

`Table[ShowGraph[Hypercube[i]], {i, 1, 4}]`

`Out[] =`



Cada una de estas representaciones como veremos, constituye un grafo. El hipercubo es un pequeño ejemplo sobre las **múltiples aplicaciones** que tienen los grafos en la **computación**, algunas de las cuales, serán abordadas en la presente unidad didáctica. Esta unidad **tiene como objetivo comprender los aspectos esenciales de la teoría de grafos**: definición, representación, trayectorias, ciclos, circuitos, algoritmos y sus aplicaciones para resolver problemas reales orientados a las ciencias de la computación. El texto recurre a un conjunto de recursos iconográficos que facilitan su consulta, la iconografía integrada es la siguiente:



Consultar video



Definición



Ejemplo



Ejercicios para el estudiante



Ponga mucha atención



Consulte el *software Mathematica*



Finalizó un ejemplo



Observación de profundización



Piense



Recuerde

Comenzaremos con algunas definiciones generales incluyendo el concepto formal de grafo.

Conceptos generales



Iniciaremos con el **concepto de grafo**.

Definición 1



Sean V y E dos conjuntos no vacíos. V un conjunto de elementos llamados nodos o vértices y E un conjunto de aristas o lados. Un grafo $G=(V,E)$ es una relación R sobre el conjunto V , tal que un nodo a está relacionado con otro b , si existe una arista $e \in E$ cuyos extremos son a y b . Si las aristas de G son dirigidas, es decir, tienen un orden en la dirección que relaciona los vértices de V , entonces a G se le llama grafo dirigido o digrafo, en caso contrario, se dice que G es un grafo no dirigido. Si G es dirigido y aRb (a y b están relacionados) al lado e de G que relaciona a y b , se le denota: $e=(a,b)$. Si G es un grafo no dirigido y aRb , la arista e que los une, se representa como: $e=[a,b]$. Dos vértices relacionados por R se llaman adyacentes. Se dice, además, que una arista e incide sobre dos nodos a y b , si e los conecta entre sí.



Un grafo de acuerdo con la definición 1, es una **relación** entre vértices vinculados por un conjunto de aristas. El estudiante recordará cómo en el tema anterior, los grafos se utilizaron para representar gráficamente una relación binaria. En general, **todo grafo es una relación binaria** que determina en el plano una figura donde los nodos se dibujan

a través de puntos primitivos y su relación, por medio de una serie de curvas que los unen.

! El *software Mathematica* cuenta con una colección de comandos fundamentales para resolver distintos tipos de problemas asociados con la teoría de grafos. Inclusive, posee el paquete *Combinatorica* introducido ya en el curso, que integra **450** instrucciones especializadas en el **área de la matemática finita** y muchos de los algoritmos clásicos a considerar en la presente unidad, ya han sido programados por medio de alguna sentencia que pertenece a este paquete. También, *Mathematica* da la posibilidad de trabajar con el tema de grafos prescindiendo de *Combinatorica*. Como el lector comprobará, *Mathematica* y *Combinatorica* presentan algunos conflictos entre sí. Pese a ello, el uso de *Mathematica* con o sin *Combinatorica*, resulta esencial para el desarrollo de los contenidos a tratar en las secciones sucesivas.

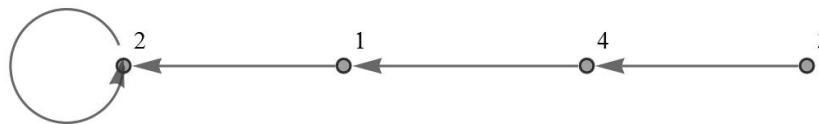


✦ Un grafo se crea en *Mathematica* por medio de las instrucciones *Graph* o *ShowGraph*. Una se diferencia de la otra por el uso del paquete *Combinatorica*. *ShowGraph* requiere de este paquete, mientras que *Graph* no lo utiliza. Consideremos el grafo $G=(V,E)$ dado por: $V = \{1,2,3,4\}$ y $E = \{(1,2),(2,2),(3,4),(4,1)\}$. Mediante *Graph* se construye así:

In[] :=

```
Graph[{1 • → 2, 2 • → 2, 3 • → 4, 4 • → 1}, VertexLabels -> "Name", ImagePadding -> 10,  
EdgeStyle -> Arrowheads[Medium], EdgeShapeFunction -> "Arrow"]
```

Out[] =



El símbolo $\bullet \rightarrow$ indica a *Mathematica* que G es un digrafo. Si G hubiese sido un

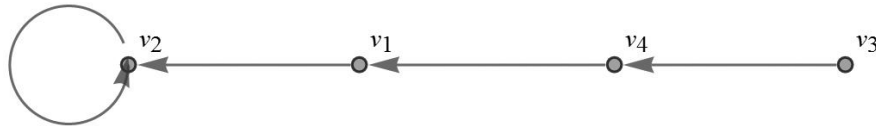
grafo no dirigido, cada $\bullet \rightarrow$ debería ser reemplazada por $\bullet \bullet$. Los comandos `DirectedEdge` y `UndirectedEdge` son equivalentes a $\bullet \rightarrow$ y $\bullet \bullet$, respectivamente. Los símbolos $\bullet \rightarrow$ y $\bullet \bullet$ se encuentran en el *software* en: *Palettes/ Special Characters/ Symbols/* \times . Además, el atributo `VertexLabels` \rightarrow "Name", acepta la sentencia `Table`. Por ejemplo:


`In[] :=`

```
Graph[{1  $\bullet \rightarrow$  2, 2  $\bullet \rightarrow$  2, 3  $\bullet \rightarrow$  4, 4  $\bullet \rightarrow$  1}, VertexLabels  $\rightarrow$  Table[i  $\rightarrow$  vi, {i, 1, 4}], ImagePadding  $\rightarrow$  10, EdgeStyle  $\rightarrow$  Arrowheads[Medium], EdgeShapeFunction  $\rightarrow$  "Arrow"]
```

(1)

`Out[] =`



 `Table` ha permitido etiquetar los nodos del grafo como v_1 , v_2 , v_3 y v_4 . Al recurrir a `ShowGraph`, se obtiene algo similar:

`In[] :=`

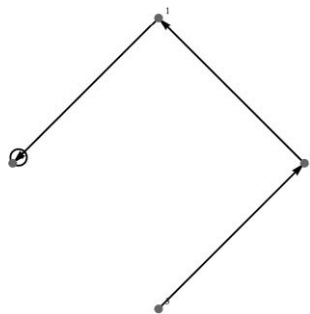
```
<< Combinatorica`
```

```
h = {{1, 2}, {2, 2}, {3, 4}, {4, 1}};
```

```
ShowGraph[G = SetGraphOptions[FromOrderedPairs[h], VertexColor  $\rightarrow$  Gray, EdgeColor  $\rightarrow$  Black], VertexLabel  $\rightarrow$  True, PlotRange  $\rightarrow$  0.1]
```

(2)

`Out[] =`



Si G fuera un grafo no dirigido, `FromOrderedPairs[h]` se sustituye por: `FromUnorderedPairs[h]`. Se advierte al alumno que al levantar el paquete `Combinatorica`, el comando `Graph` no es reconocido por el *software*. Una solución a este problema consiste en cerrar la sesión del *kernel* de *Mathematica* toda vez que se levante este paquete. Lo anterior se puede ejecutar de dos maneras:

- Manualmente en el menú: *Evaluation/ Quit Kernel/ Local*.
- De forma automática por medio del comando: `Quit[]`.



Se sugiere al lector siempre que se levante el paquete `Combinatorica`, finalizar el recorrido de las instrucciones con `Quit[]`. Esto evitará errores de ejecución por parte de *Mathematica* dado el conflicto existente con `Combinatorica`. Como veremos, algunos comandos requerirán el uso de `Combinatorica` y otros no, por lo que no siempre será necesario cerrar la sesión del *kernel*.



Es importante destacar otra diferencia entre `Graph` y `ShowGraph`. `Graph` acepta como nodos cualquier lista de vértices sean estos números reales o letras del abecedario. `ShowGraph`, en cambio, toma como vértices de un grafo $G=(V,E)$, **únicamente números naturales consecutivos** $\{1,2,\dots,n\}$, con n la cardinalidad de V .

Cuando se usa el comando `Graph` para crear un grafo $G=(V,E)$, las instrucciones `VertexList` y `EdgeList` devuelven los conjuntos V y E . Por ejemplo, para el grafo expuesto en (1):

`In[] :=`

`G = Graph[{1 • → 2, 2 • → 3, 3 • → 4, 4 • → 1}, VertexLabels -> Table[i -> Subscript[v, i], {i, 1, 4}],`

`ImagePadding -> 10, EdgeStyle -> Arrowheads[Medium], EdgeShapeFunction -> "Arrow"];`


`VertexList[G]`

`EdgeList[G]`

`Out[] =`

{1, 2, 3, 4}

{1• → 2, 2• → 2, 3• → 4, 4• → 1}

 Como se observa, la lista de vértices no debe ser confundida con sus etiquetas o *labels*. La sentencia `Edges` del paquete `Combinatorica`, al igual que `EdgeList`, retorna la lista de lados de un grafo $G = (V, E)$. Con relación a la lista de vértices, `Combinatorica` tiene la instrucción `Vertices` que retorna las coordenadas en el plano cartesiano de los nodos $\{1, 2, \dots, n\}$, siendo n la cardinalidad de V . Tomaremos para ejemplificar el uso de estos comandos, el grafo mostrado en (2):

`In[] :=`

```
<< Combinatorica`
```

```
h = {{1, 2}, {2, 2}, {3, 4}, {4, 1}};
```

```
ShowGraph[G = SetGraphOptions[FromOrderedPairs[h], VertexColor -> Gray, EdgeColor -> Black], VertexLabel -> True, PlotRange -> 0.1];
```

```
Edges[G]
```


```
Vertices[G]
```

```
Quit[]
```

`Out[] =`

```
{{1, 2}, {2, 2}, {3, 4}, {4, 1}}
```

```
{{0, 1.}, {-1., 0}, {0, -1.}, {1., 0}}
```

 Si en un grafo $G = (V, E)$ una arista e conecta a un nodo a consigo mismo, a dicha arista se le denomina **lazo o ciclo** de G . En el nodo v_2 del grafo (2), ocurre un lazo. En este contexto, *Mathematica* posee el comando `SimpleGraphQ`. Éste forma una función booleana que retorna `True` si el grafo G que entra como argumento, no tiene lazos y `False`, en otro caso. `SimpleGraphQ` debe utilizarse junto con `Graph`. Un grafo creado con `ShowGraph` no es interpretado adecuadamente por esta sentencia.

Otro concepto relacionado con las aristas de un grafo consiste en la definición de lados múltiples. Se dice que dos nodos a y b poseen lados múltiples o paralelos en un grafo G , si existe más de una arista entre a y b . El grafo no dirigido mostrado a continuación, posee lados múltiples entre los vértices 1 y 2:

`In[] :=`

`<<Combinatorica``

`h = {{1, 2}, {1, 2}, {1, 2}, {3, 1}, {3, 4}};`

`ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Gray, EdgeColor -> Black], VertexLabel -> True, PlotRange -> 0.1]`

`Quit[]`

(3)

`Out[] =`



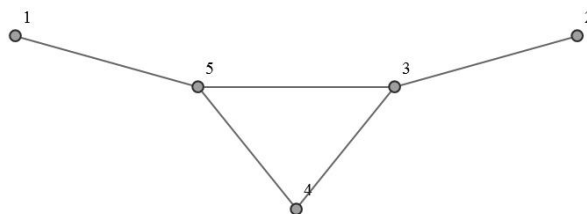
✳ Tanto `Graph` como `ShowGraph` soportan grafos con lados múltiples. Un grafo G sin lazos y sin lados múltiples recibe el nombre de grafo simple. Los grafos compartidos en (1) y (3) no son simples. El grafo G mostrado a continuación, es simple:

`In[] :=`

`RandomGraph[{5, 5}, VertexLabels -> "Name", ImagePadding -> 10]`

(4)

`Out[] =`



El comando `RandomGraph` construye si es posible, un grafo simple G , pseudoaleatorio con n vértices y m aristas. En el ejemplo anterior, se ha especificado para G , 5 nodos y 5 lados.



Una ruta, camino o trayectoria entre dos nodos a y b de un grafo $G=(V,E)$ es una **sucesión de vértices** de V que conecta al nodo a con el vértice b , por medio de las aristas de E sobre G , **sin repetición** de lados. La longitud de la ruta se define como la **cantidad de aristas** de G utilizadas para conectar a con b . En (4) una ruta de 1 a 2 está dada por: $1 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2$ y es de longitud 4. En general, si se usan t nodos en una trayectoria, su longitud corresponde a $t-1$. *Mathematica* encuentra rutas entre dos nodos por medio del comando `NetworkFlow`. El siguiente código muestra las salidas obtenidas, sobre un grafo creado con `ShowGraph`:

```
In[ ] :=
```

```
<< Combinatorica`
```

```
h = {{1, 2}, {1, 3}, {2, 4}, {3, 2}, {3, 4}, {4, 5}, {3, 5}};
```

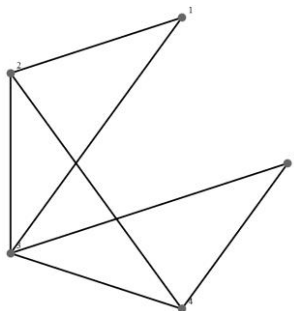
```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Gray,
EdgeColor -> Black], VertexLabel -> True, PlotRange -> 0.1]
```

```
NetworkFlow[G, 1, 5]
```

```
NetworkFlow[G, 1, 5, Edge]
```

```
Quit[]
```

```
Out[ ] =
```




```
2
```

```
{{{1, 2}, 1}, {{1, 3}, 1}, {{2, 4}, 1}, {{3, 5}, 1}, {{4, 5}, 1}}
```

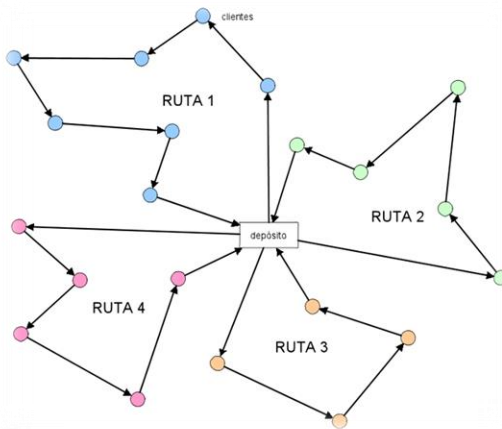
El 2 recibido de $\text{NetworkFlow}[G, 1, 5]$ representa la cantidad máxima de caminos de 1 a 5 que el *software* determinó y la lista subsiguiente generada por $\text{NetworkFlow}[G, 1, 5, \text{Edge}]$, construye las posibles trayectorias. En este ejemplo, se concluye por interpretación, que dos rutas de 1 a 5 son: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ y $1 \rightarrow 3 \rightarrow 5$. El siguiente programa automatiza la elaboración de las rutas:

```
In[ ] :=
<<Combinatorica`
h = {{1, 2}, {1, 3}, {2, 4}, {3, 2}, {3, 4}, {4, 5}, {3, 5}};
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Gray,
EdgeColor -> Black], VertexLabel -> True, PlotRange -> 0.1];
CantiRutas = NetworkFlow[G, 1, 5];
L = NetworkFlow[G, 1, 5, Edge];
For[i = 1, i <= CantiRutas, v = L[[i, 1]]; camino = ToString[v[[1]] -> v[[2]]; elemento = v[[2]]; For[j
= i + 1, j <= Length[L], w = L[[j, 1]]; If[elemento == w[[1]], camino = StringJoin[camino, " -> ",
ToString[w[[2]]]]; elemento = w[[2]]; j++]; Print[ToExpression[camino]]; i++]
Quit[]
Out[ ] =
1 -> 2 -> 4 -> 5
1 -> 3 -> 5
```

 Además de las instrucciones propias del *software Mathematica* se contará para efectos de la presente materia con el patrocinio de un **paquete denominado**: VilCretas. VilCretas se ha desarrollado con la principal intención de trabajar desde un punto de vista **educativo** en el campo de la **matemática discreta**, utilizando como apoyo el *software* comercial *Mathematica*. El paquete está constituido por una serie de funciones que por defecto no están integradas en *Mathematica*, abarcando las principales áreas de contenido de un curso clásico de estructuras discretas. Puede ser descargado libremente desde la dirección: <http://www.escinf.una.ac.cr/discretas/index.php/archivos/category/7-packages> y una explicación sobre la forma en cómo se instala se presenta en:



Otra tarea interesante, consiste en **contar el número de rutas posibles** sobre un grafo simple. Estas trayectorias se pueden determinar por medio de la operación composición de relaciones. Si recordamos, la definición 1, propone que un grafo G es una relación R entre sus vértices. Al tomar la composición RoR , se forman un conjunto



de pares cuyas componentes, excluyendo aquellos donde la primera y la segunda coordenada coinciden, son los extremos de todas las rutas de longitud 2 en G . Si se compone ahora, $Ro(RoR)$ el resultado muestra los extremos de todos los caminos de longitud 3, eliminando aquellos donde la primera componente es igual a la segunda. Si se calcula posteriormente, $Ro[Ro(RoR)]$ se forman los

extremos de todas las rutas de longitud cuatro, omitiendo los pares ordenados donde las coordenadas son iguales y así sucesivamente. Aunque la lógica es interesante, plantea el problema de incluir en cada lista, rutas con repetición de aristas. Utilizando el *software Mathematica*:

<< Combinatorica`

```
CantRutas[G_Graph, n_] := Module[{Inversos, Eliminalguales, Composiciones},
Composiciones[R_List, H_List] := Module[{Composicion = {}}, For[i = 1, i <= Length[H], For[j = 1, j <= Length[R], If[H[[i, 2]] == R[[j, 1]], Composicion = Append[Composicion, {H[[i, 1]], R[[j, 2]]}]; j++]; i++]; Return[DeleteDuplicates[Composicion]]; Inversos[R_List] := Module[{NR = {}}, For[i = 1, i <= Length[R], NR = Append[NR, {R[[i, 2]], R[[i, 1]]}]; i++]; Return[NR]; Eliminalguales[R_List] := Module[{NR = R}, For[i = 1, i <= Length[NR], If[NR[[i, 1]] == NR[[i, 2]], NR = Delete[NR, i]; i++]; Return[NR]; R = Union[Edges[G], Inversos[Edges[G]]]; c = R; For[i =
```

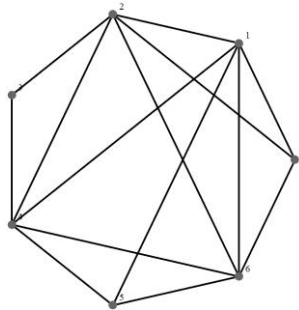
```
1, i <= n - 1, c = Composiciones[R, c]; c = Eliminalguales[c]; i++; Print["El grafo tiene ",
Length[c], " rutas de longitud ", n, ", con extremos: "]; Return[Sort[c]]] (5)
```

La función CantRutas entrega como salida:

El grafo tiene 42 rutas de longitud 20, con extremos:

```
{{1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {1, 7}, {2, 1}, {2, 3}, {2, 4}, {2, 5}, {2, 6}, {2, 7}, {3, 1}, {3, 2}, {3, 4}, {3,
5}, {3, 6}, {3, 7}, {4, 1}, {4, 2}, {4, 3}, {4, 5}, {4, 6}, {4, 7}, {5, 1}, {5, 2}, {5, 3}, {5, 4}, {5, 6}, {5, 7}, {6, 1},
{6, 2}, {6, 3}, {6, 4}, {6, 5}, {6, 7}, {7, 1}, {7, 2}, {7, 3}, {7, 4}, {7, 5}, {7, 6}}
```

Tomando como ejemplo $n = 20$ y el grafo G :



✨ En el paquete VilCretas el estudiante puede consultar las instrucciones Ruta y CantRutas que resuelven el problema de la identificación de rutas y el número de caminos, respectivamente, entre cualquier par de vértices. Para ello es posible observar:



📖 Un grafo G se dice que es conexo, cuando entre cada par de nodos cualesquiera siempre **existe una ruta**. En caso contrario, se dice que el grafo es desconexo. La instrucción ConnectedGraphQ determina en *Mathematica* si dado un grafo G , este es conexo. Por ejemplo, el código próximo, comprueba la conjetura “**un grafo no dirigido**

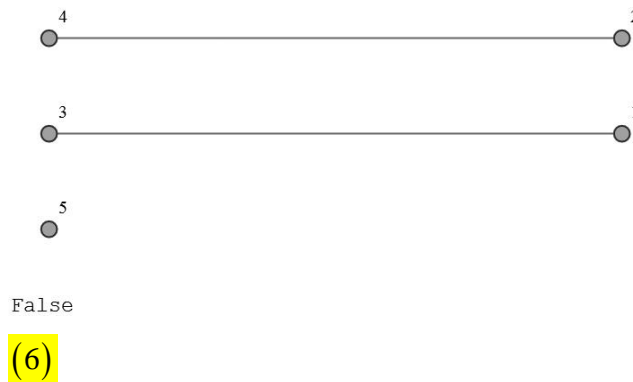
con cinco nodos y dos aristas siempre es desconexo”:

`In[] :=`

`G = RandomGraph[{5, 2}, VertexLabels -> "Name", ImagePadding -> 10]`

`ConnectedGraphQ[G]`

`Out[] =`



Un nodo sin aristas incidentes se denomina **aislado**. El grafo de la figura (6), posee el nodo aislado 5.

Los grafos no dirigidos se subdividen en un conjunto de partes llamadas **piezas o componentes**. Estas se derivan de una relación de equivalencia. Consideremos el siguiente teorema.

Teorema 1

Sea $G = (V, E)$ un grafo no dirigido, entonces la relación R' definida sobre V , tal que: $\forall a, b \in V, aR'b$ sí y solo sí existe una trayectoria que une a con b , es una relación de equivalencia. Las clases de equivalencia de R' se llaman componentes o piezas de G .



Estas **clases de equivalencia** reciben el nombre de piezas del grafo, pues forman una serie de subgrafos conexos que no tienen nada en común. Es decir, las clases de equivalencia de R' , son los subgrafos conexos en los que se puede **subdividir** el grafo original. Naturalmente, si G es conexo, la relación de equivalencia establecida en el

teorema 1, produce una única clase de equivalencia, que corresponde al mismo grafo.

Ejemplo 1



Verifique el teorema 1 por medio de un grafo pseudoaleatorio generado en *Mathematica*.

Mathematica contiene el comando `ConnectedComponents`, capaz de crear la lista de nodos que representan todas las componentes o piezas de un grafo no dirigido. En este sentido, la solución expuesta, construye la relación R' por medio de `ConnectedComponents` e invoca a la función `TipoRelacion` que forma parte del paquete de *software* `VilCretas`:

```
In[ ] :=
```

```
(* Genera un grafo pseudoaleatorio G y crea R' *)
```

```
G = RandomGraph[{20, 18}, VertexLabels -> "Name", ImagePadding -> 10]
```

```
A = VertexList[G]; MC = ConnectedComponents[G];
```

```
<< Combinatorica`
```

```
R = {};
```

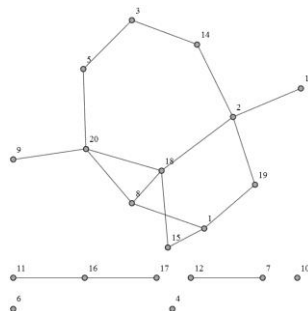
```
For[i = 1, i <= Length[MC], L = CartesianProduct[MC[[i]], MC[[i]]]; R = Union[L, R]; i++];
```

```
Print["R' es:"];
```

```
<< VilCretas`
```

```
TipoRelacion[R, A]; Quit[]
```

```
Out[ ] =
```



(7)

R' es:

La relación es reflexiva

La relación es simétrica

La relación no es antisimétrica

La relación es transitiva

La relación es de equivalencia



El alumno al correr cada vez el código anterior obtiene un grafo pseudoaleatorio distinto sobre el cuál se verifica el teorema 1. Se sugiere al estudiante continuar experimentando al aumentar o disminuir la cantidad de nodos y aristas en RandomGraph. Por defecto, las respuestas mostradas en (7), toman 20 vértices y 18 lados ¡Sería interesante ejemplificar a través de *software*, la falsedad del teorema 1 si G es un digrafo! Para más detalles sobre el empleo del comando TipoRelacion se puede abrir:



En ciertos tipos de problemas a cada una de las aristas que determinan un grafo se les suele **asociar** un **número real**, usualmente no negativo. A estos números se les denomina **pesos** de los lados y en dicho caso, se dice que el grafo es **ponderado**. En *Mathematica* un grafo ponderado se crea asignando pesos a sus aristas. Usando Graph, se tendría:

```
G = Graph[{1 ↔ 2, 2 ↔ 3, 3 ↔ 1}];
```

```
G = Graph[{1 ↔ 2, 2 ↔ 3, 3 ↔ 1}, EdgeWeight -> RandomReal[{1, 5}, EdgeCount[G]]] (8)
```



EdgeWeight -> RandomReal[{1, 5}, EdgeCount[G]] asigna a cada lado del grafo G , un número real pseudoaleatorio mayor o igual a 1 y menor o igual a 5. Si se desea conocer

con precisión los pesos de cada una de las aristas de G , se usa la instrucción `WeightedAdjacencyMatrix[G]`. Esta devuelve una matriz cuyas filas y columnas representan los nodos del grafo, donde la entrada (i, j) es el peso de la arista que une el vértice v_i con el nodo v_j . A esta matriz se le llama **matriz de adyacencia de pesos** sobre G . En (8) retorna:

`In[] :=`

`MatrixForm[WeightedAdjacencyMatrix[G]]`

`Out[] =`

$$\begin{pmatrix} 0 & 3.09382 & 1.4105 \\ 3.09382 & 0 & 4.98849 \\ 1.4105 & 4.98849 & 0 \end{pmatrix}$$


! El lector ya habrá previsto que la salida mostrada en esta ejecución es **seudoaleatoria**, por lo tanto, el resultado variará de un recorrido a otro.

✦ Para la creación de grafos en el *software Mathematica* se sugiere, además, consultar el comando `GrafoC` del paquete `VilCretas`, en:



✦ Además, la instrucción `PesosAristas` del paquete `VilCretas` también **retorna** los pesos vinculados a los lados de un grafo ponderado. Si se desea obtener más detalles sobre ella, se puede visualizar:



 *Mathematica* tiene el comando `WeightedGraphQ[G]` función booleana con valor `True` si G es un grafo ponderado y `False`, si los lados de G no poseen pesos. Esta sentencia **no funciona** si el grafo se ha creado con `ShowGraph`.

También, un grafo G ponderado se puede implementar usando `ShowGraph`, veamos:

```
In[ ] :=
```

```
<< Combinatorica`
```

```
h = {{1, 2}, {2, 3}, {3, 1}};
```

```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Gray,  
EdgeColor -> Black], VertexLabel -> True, PlotRange -> 0.1];
```


```
G = SetEdgeWeights[G, RandomReal[{1, 5}, M[G]]];
```

```
GetEdgeWeights[G]
```

```
Quit[]
```

```
Out[ ] =
```

```
{1.86468, 4.37716, 2.38587}
```

 La sentencia `SetEdgeWeights` asigna los pesos a las aristas de G y `GetEdgeWeights` devuelve la lista de pesos de cada uno de los lados de G . Esta instrucción permite **detallar** las aristas para las cuales se desea obtener su peso, por ejemplo, suponga que se requiere el peso de los lados $[1,2]$ y $[3,1]$ únicamente, entonces se especifica dentro de `GetEdgeWeights`, así: `GetEdgeWeights[G, {{1, 2}, {3, 1}}]`. Además, es destacable que el comando `M[G]` encuentra el **número de aristas** del grafo. Aunque no se utiliza en este ejemplo, algo similar lleva a cabo la instrucción `V[G]`, contando la **cantidad de nodos** que existen sobre G .



En un grafo ponderado la longitud de una trayectoria se define como la **suma** de los pesos de las aristas involucradas en la ruta. `CostOfPath` es un comando de *Mathematica* que retorna la **longitud de un camino** sobre un grafo G , dada la lista de nodos que determinan la trayectoria. Si la lista forma un camino no existente en G , el comando retorna ∞ .

Ejemplo 2



Encuentre la longitud de los caminos construidos por `NetworkFlow` que unen los vértices 1 y 6, en el grafo:

```
<< Combinatorica`
```

```
h = {{1, 2}, {1, 6}, {2, 3}, {3, 4}, {2, 4}, {5, 1}, {5, 6}, {6, 4}};
```

```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Gray,
EdgeColor -> Black], VertexLabel -> True, PlotRange -> 0.1]
```

```
G = SetEdgeWeights[G, RandomReal[{1, 5}, M[G]]];
```

En *Mathematica*:

```
In[ ] :=
```

```
<< Combinatorica`
```

```
h = {{1, 2}, {1, 6}, {2, 3}, {3, 4}, {2, 4}, {5, 1}, {5, 6}, {6, 4}};
```

```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Gray,
EdgeColor -> Black], VertexLabel -> True, PlotRange -> 0.1]
```

```
CantiRutas = NetworkFlow[G, 1, 6];
```

```
L = NetworkFlow[G, 1, 6, Edge];
```

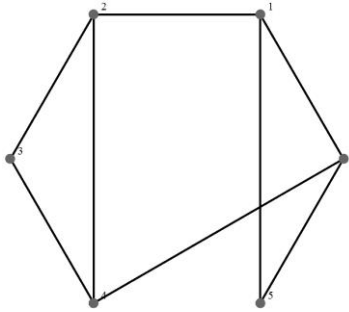
```
G = SetEdgeWeights[G, RandomReal[{1, 5}, M[G]]];
```

```
For[i = 1, i <= CantiRutas, v = L[[i, 1]]; Ti = {v[[1]], v[[2]]}; elemento = v[[2]]; For[j = i + 1, j <=
Length[L], w = L[[j, 1]]; If[elemento == w[[1]], Ti = Append[Ti, w[[2]]]; elemento = w[[2]]; j++];
i++]
```

```
For[i = 1, i <= CantiRutas, Print[Ti, " tiene longitud: ", CostOfPath[G, Ti]]; i++]
```

```
Quit[]
```

```
Out[ ] =
```



{1, 2, 4, 6} tiene longitud: 10.8441

{1, 5, 6} tiene longitud: 5.92865

{1, 6} tiene longitud: 1.13457



🌟 **GraphDistance** es una instrucción similar a **CostOfPath**. En el *software*, **GraphDistance**[G, a, b] calcula la **distancia más corta** entre los nodos a y b . Si se escribe: **GraphDistance**[G, a], encuentra además, las distancias más pequeñas que unen el nodo a con **todos** los vértices del grafo G . Este comando no es compatible con el paquete **Combinatorica**.

Otro concepto primordial en teoría de grafos estriba en la definición de **valencia**.


Definición 2

📖 Sea $G=(V,E)$ un grafo. Si G es no dirigido y $a \in V$, se llama **valencia o grado del nodo a** , al número de lados de E que son incidentes sobre a , este se representa como $\theta(a)$. Si en a ocurre un lazo, el ciclo suma 2 unidades a su grado. Por otra parte, si G es dirigido, se llama **valencia o grado interno de a** , denotado $\theta_*(a)$ al número de aristas de G que llegan al vértice a . La cantidad de lados de G que salen del nodo a , recibe el nombre de **valencia o grado externo de a** y se representa así: $\theta^*(a)$.

Las valencias o grados de los vértices de un grafo son muy importantes pues tienen diversas aplicaciones en el contexto de esta teoría, una de ellas se verá más adelante en el tema de circuitos. En un grafo no dirigido, la **suma de las valencias** da como

resultado un número que depende de la **cantidad de lados** del grafo. El siguiente teorema garantiza esta propiedad.

Teorema 2

 Sea $G=(V,E)$ un grafo no dirigido con $V = \{v_1, v_2, \dots, v_n\}$ y m su número de aristas, entonces:

$$\sum_{i=1}^n \theta(v_i) = 2m$$

Ejemplo 3



Elabore una rutina en Mathematica que verifique el teorema 2.

La solución se presentará de dos maneras distintas. En la primera, se crean 10 grafos pseudoaleatorios con tamaños diversos mediante un ciclo For, donde en cada iteración se comprueba el teorema. En la segunda, usando la sentencia UniformGraphDistribution dentro de RandomGraph, se producen m grafos pseudoaleatorios con n vértices y n lados. Los valores de n y m se leen por teclado.

- Solución 1:

```
In[ ] :=
```

```
For[i = 2, i <= 11, Print[G = RandomGraph[{2*i, 2*i}, VertexLabels -> "Name", ImagePadding -> 10]]; Print["El valor de verdad de la propiedad en este grafo es: "]; If[Total[VertexDegree[G]] == 2 EdgeCount[G], Print["True"], Print["False"]]; i++]
```



El `Out[]` muestra cada grafo construido y el valor de verdad de la comparación entre la suma de las valencias y el doble de la cantidad de lados. En todos los casos la respuesta es True, lo cual corrobora el enunciado del teorema 2.

- Solución 2:

```
In[ ] :=
```

```
n = Input["Digite la cantidad de nodos y aristas de los grafos:"];  
m = Input["Digite la cantidad de grafos a generar:"];
```





```
LG = RandomGraph[UniformGraphDistribution[n, n], m, VertexLabels -> "Name",  
ImagePadding -> 10];
```

```
For[i = 1, i <= Length[LG], G = LG[[i]]; Print[G]; Print["El valor de verdad de la propiedad en este  
grafo es: "]; If[Total[VertexDegree[G]] == 2 EdgeCount[G], Print["True"], Print["False"]]; i++]
```

! Como se observa en los programas anteriores, `VertexDegree` encuentra un vector que contiene las valencias de todos los nodos de un grafo G . *Mathematica* posee también, el comando `Degrees` del paquete `Combinatorica`. `Degrees` devuelve al igual que `VertexDegree`, el vector de grados de los vértices de un grafo.



 Se previene al lector que la sentencia `Degrees` suma **una unidad por cada lazo** en la valencia de un nodo que lo contiene y **no dos unidades**. Según la definición 2, un lazo agrega dos unidades al grado, por consiguiente, si se usa esta instrucción en un grafo con lazos, se debe añadir una unidad más a la valencia de aquellos vértices que los posean. `VertexDegree` no presenta este inconveniente.

 En un grafo G dirigido, el *software* cuenta con las instrucciones: `VertexInDegree` y `VertexOutDegree`. Estas determinan las valencias internas y externas de todos los nodos de G , respectivamente. Además, las sentencias del paquete `Combinatorica`: `InDegree` y `OutDegree`, construyen de la misma forma, las listas de grados internos y externos.

También, el comando `Valencias` del paquete `VilCretas` calcula los grados de los nodos de un grafo cualesquiera. Para más detalles sobre esta instrucción se puede visualizar:



Ejemplo 4



Conjeture qué ocurre con el teorema 2 si el grafo es dirigido.

El comando `RandomGraph` acepta el atributo `DirectedEdges -> True`, que crea en *Mathematica* grafos pseudoaleatorios dirigidos.

El alumno podría intuir que la suma de las valencias internas y externas en un digrafo, se espera, dé como resultado el **doblo** de la cantidad de aristas. Asimismo, al experimentar sobre algunos casos particulares, se descubre otra interesante propiedad, al parecer, la suma de las valencias internas es **igual** a la suma de los grados externos. El código mostrado verifica estas dos hipótesis:

```
In[ ] :=
```

```
n = Input["Digite la cantidad de nodos y aristas de los grafos:"];
```

```
m = Input["Digite la cantidad de grafos a generar:"];
```

```
LG = RandomGraph[UniformGraphDistribution[n, n], m, DirectedEdges -> True,  
VertexLabels -> "Name", ImagePadding -> 10];
```

```
For[i = 1, i <= Length[LG], G = LG[[i]]; Print[G]; Print["El valor de verdad de la propiedad en este  
grafo es: "]; If[(Total[VertexInDegree[G]] == Total[VertexOutDegree[G]]) &&  
(Total[VertexInDegree[G]] + Total[VertexOutDegree[G]] == 2 EdgeCount[G]), Print["True"],  
Print["False"]]; i++]
```



En todos los digrafos, el estudiante comprobará que el valor de verdad devuelto por el *software* es `True`. Se concluye entonces la conjetura: “**en un grafo dirigido sin lazos ni lados múltiples, la suma de los grados internos y externos es el doble de la cantidad de lados y a su vez, la suma de las valencias internas es igual a la suma de los grados externos**”. Esta propiedad es falsa si el grafo no es simple.



El concepto de **excentricidad** es otra definición importante.

Definición 3



Sea $G=(V,E)$ un grafo conexo y $a \in V$. Se llama *excentricidad del vértice a* , al valor máximo de las longitudes mínimas de los caminos desde a a los demás nodos de G . Al valor mínimo de todas las excentricidades en G , se le denomina *radio del grafo*. Al subgrafo que contiene los vértices donde ocurre la excentricidad mínima, se le define como el *centro de G* . Por otra parte, al valor máximo de las excentricidades se le llama *diámetro del grafo*.



La excentricidad de un vértice en un grafo conexo es por la definición anterior, la **mayor longitud** de las rutas más cortas entre el nodo y cualquier otro vértice del grafo. Si G es ponderado la excentricidad se obtiene por medio de los **pesos asignados** a cada una de sus aristas.

Ejemplo 5



Utilizando el software Mathematica, encuentre el centro, el radio y el diámetro de distintos grafos conexos pseudoaleatorios.

Las instrucciones que prosiguen resuelven lo solicitado:

```
In[ ] :=
```

```
GrafoSAConexo[n_] := Module[{}, G = RandomGraph[{n, n}, VertexLabels -> "Name",  
ImagePadding -> 10]; While[ConnectedGraphQ[G] == False, G = RandomGraph[{n, n},  
VertexLabels -> "Name", ImagePadding -> 10]]; Return[G]]; For[i = 1, i <= 10, LE = {}; Centro =  
{}; Print[G = GrafoSAConexo[20]]; For[j = 1, j <= 20, LE = Append[LE, VertexEccentricity[G, j]];  
j++]; vmin = Min[LE]; vmax = Max[LE]; For[h = 1, h <= Length[LE], If[vmin == LE[[h]], Centro =  
Append[Centro, h]]; h++]; Print["El centro del grafo anterior es el subgrafo con vértices ",  
Centro, " su radio es ", vmin, " y su diámetro corresponde a ", vmax]; i++]
```

(9)

GrafoSAConexo es una función creada para retornar grafos conexos pseudoaleatorios producidos por RandomGraph. Además, el comando VertexEccentricity[G, j] determina la excentricidad de cada nodo j sobre G .



El centro, radio y diámetro de un grafo conexo se puede encontrar más directamente usando los comandos: `GraphRadius`, `GraphDiameter` y `GraphCenter`, respectivamente. Cada uno de ellos, recibe como parámetro el grafo G . Se insta al alumno a modificar (9), recurriendo a estas instrucciones.



Los grafos se pueden **clasificar** en distintos tipos que serán abordados en la siguiente sección.

Tipos de grafos



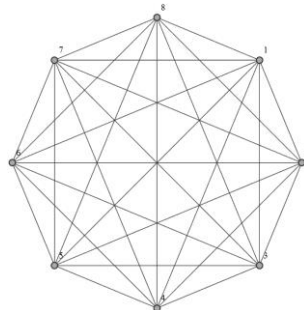
Mathematica y el paquete *VilCretas* poseen múltiples comandos que construyen **tipos especiales de grafos**. Veamos algunos de ellos.

`CompleteGraph`: genera un **grafo completo**. Se entiende un grafo completo como aquel grafo simple, donde **para cualquier par** de vértices siempre existe una arista que los une. Se suele denotar como K_n , siendo n la cantidad de nodos. Por ejemplo, la siguiente sentencia forma el grafo completo K_8 :

```
In[ ] :=
```

```
CompleteGraph[8, VertexLabels -> "Name", ImagePadding -> 10]
```


```
Out[ ] =
```



Los grafos completos satisfacen tener un número m de aristas que **dependen**

directamente de la cantidad n de vértices. Se puede comprobar que:

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

 Dejaremos esto como un ejercicio para el lector. Es necesario, también recalcar, que *Mathematica* posee el comando `CompleteQ` del paquete `Combinatorica`. Esta instrucción se usa para determinar si un grafo es completo. Por ejemplo, las siguientes líneas de código construyen K_4 y `CompleteQ` retorna `True`:

```
In[ ] :=
```

```
<< Combinatorica`
```

```
h = {{1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}};
```


```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black,  
EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1]
```

```
CompleteQ[G]
```

```
Quit[]
```

```
Out[ ] =
```

```
True
```

 La instrucción `GrafoCompleto` de `VilCretas` facilita también la construcción de este tipo de grafos. Para más información se puede consultar:



`EmptyGraph`: se llama **grafo discreto**. Cumple con la condición de **no contener** ninguna arista. Se le representa como D_n , siendo n el número de vértices que integra. El comando usa el paquete `Combinatorica`. `EmptyGraph[8]` construye el grafo discreto de orden 8:

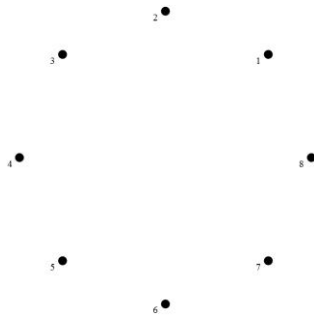
```
In[ ] :=
```

```
<< Combinatorica`
```

```
ShowGraph[EmptyGraph[8], VertexNumber -> True, PlotRange -> 0.1]
```

```
Quit[]
```

```
Out[ ] =
```



✶ Adicionalmente la sentencia `EmptyQ[G]` de `Combinatorica`, verifica si el grafo G es discreto devolviendo el valor lógico `True`, o `False` si G incluye algún lado.

El comando `GrafoVacio` del paquete `VilCretas` resuelve, además, la construcción de este tipo de grafos. El alumno puede visualizar:



`CompleteKPartiteGraph`: construye un **grafo bipartito completo**. El comando usa el paquete `Combinatorica`. Este tipo de grafos simples son aquellos formados por dos conjuntos **disjuntos** de vértices y todas las posibles aristas que los unen. El grafo bipartito con n y m nodos disjuntos, se representa por: $K_{n,m}$. A continuación, se muestra el grafo bipartito completo $K_{2,3}$:

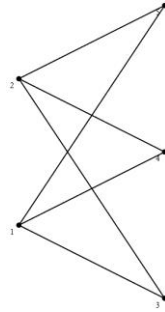
```
In[ ] :=
```


<< Combinatorica`

```
ShowGraph[CompleteKPartiteGraph[2, 3], VertexNumber -> True, PlotRange -> 0.1]
```

```
Quit[]
```

Out[] =



 Al igual que en los grafos completos, los grafos bipartitos completos, poseen una **fórmula** para la cantidad h de aristas. La rutina expuesta a continuación, calcula el **número de lados** en un grafo bipartito completo $K_{i,i+1}$, con $2 \leq i \leq 10$:

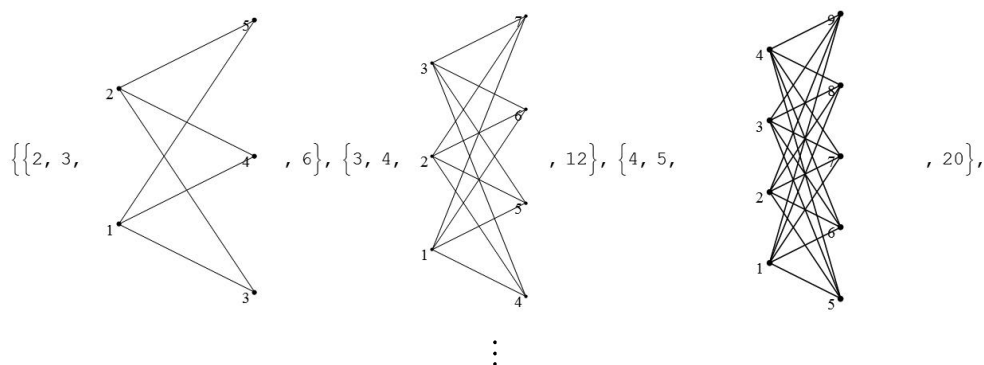
In[] :=

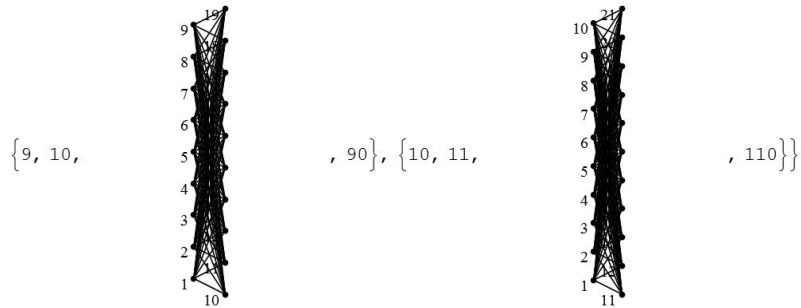
<< Combinatorica`

```
Table[{i, i + 1, ShowGraph[G = CompleteKPartiteGraph[i, i + 1], VertexNumber -> True, PlotRange -> 0.1], Length[Edges[G]]}, {i, 2, 10}]
```

```
Quit[]
```

Out[] =





! De estos resultados se conjetura que: $h = n \cdot m$ en $K_{n,m}$. *Mathematica* incorpora, además, el comando `BipartiteGraphQ`, función booleana con valor lógico `True` si su argumento es un grafo bipartito y `False`, si ocurre lo contrario. `BipartiteGraphQ` no usa el paquete `Combinatorica` y se aplica en cualquier tipo de grafo bipartito, sea este completo o no. Un grafo bipartito incompleto, tiene la misma forma de un grafo bipartito completo, **sin algunas de sus aristas**.



La instrucción `GrafoBipartitoCompleto` del paquete `VilCretas` crea grafos de este tipo, ¿qué diferencias se visualizan con respecto al comando `CompleteKPartiteGraph`? Se recomienda consultar:

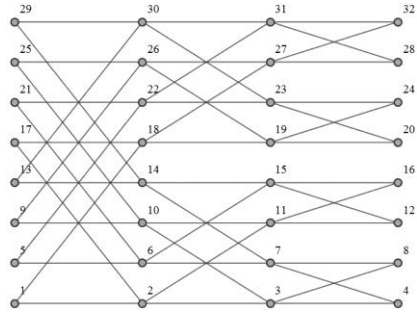



`ButterflyGraph`: se denomina **grafo mariposa**. Por ejemplo, el grafo mariposa de orden 3 se crea en *Mathematica* así:

`In[] :=`

`ButterflyGraph[3, VertexLabels -> "Name", ImagePadding -> 10]`

`Out[] =`



 La instrucción **GrafoMariposa** del paquete **VilCretas** representa también este tipo de grafos. Para indagar su funcionamiento es posible consultar:

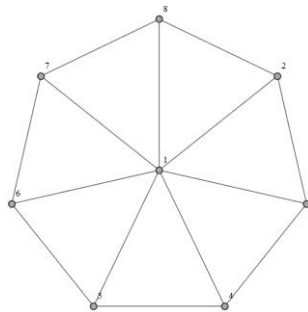



WheelGraph: llamado **grafo rueda**. Es un grafo semejante a la forma que tiene una rueda en una *carreta de bueyes*. El grafo rueda de orden 8 es:

In[] :=

`WheelGraph[8, VertexLabels -> "Name", ImagePadding -> 10]`

Out[] =



 El comando **GrafoRueda** de **VilCretas** representa, además, este tipo de grafos. Se sugiere observar:

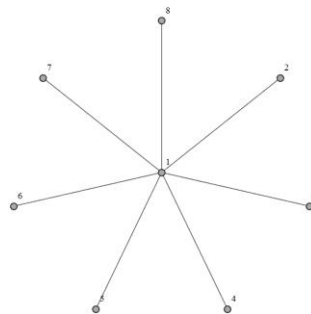



StarGraph: genera la figura de una **estrella**. Intuitivamente, el comando devuelve el grafo rueda sin las aristas externas. Por ejemplo, el grafo estrella de orden 8, tiene la forma:

In[] :=

```
StarGraph[8, VertexLabels -> "Name", ImagePadding -> 10]
```

Out[] =



 La sentencia **GrafoEstrella** del paquete **VilCretas** resuelve este tipo de gráfica. Se recomienda ver:

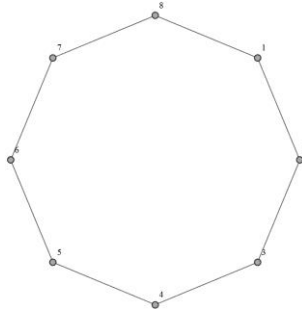



CycleGraph: se denomina **grafo círculo**. Retorna casi el mismo grafo de **WheelGraph**, exceptuando las aristas internas. Por ejemplo, a continuación, se muestra el grafo círculo de orden 8:

In[] :=

```
CycleGraph[8, VertexLabels -> "Name", ImagePadding -> 10]
```

Out[] =



 En VilCretas se encuentra disponible el comando `GrafoCiclo` para efectuar este tipo de gráfica. Con estos fines, se puede consultar:

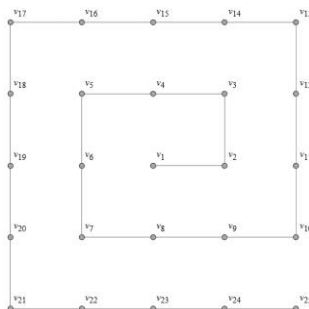


`PathGraph`: llamado **grafo camino**, interconecta una serie de vértices de manera consecutiva. El siguiente código construye un grafo camino con 25 nodos:

In[] :=

```
PathGraph[Table[i, {i, 1, 25}], VertexLabels -> Table[i -> Subscript[v, i], {i, 1, 25}],  
ImagePadding -> 10]
```

Out[] =



✦ La instrucción `GrafoCamino` del paquete `VilCretas` construye también este tipo de grafos. Para profundizar se puede observar:

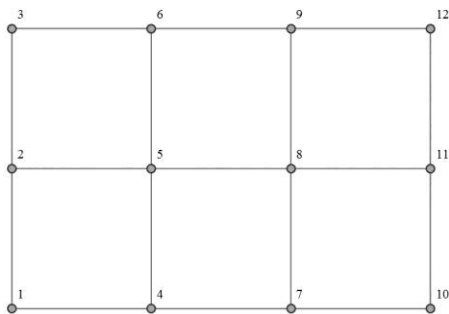


`GridGraph`: denominado **grafo cuadrícula**, forma en el plano una cuadrícula con n filas, m columnas y $n \cdot m$ nodos. Por ejemplo:

`In[] :=`

```
GridGraph[{3, 4}, VertexLabels -> "Name", ImagePadding -> 10]
```

`Out[] =`



✦ `GrafoRed` es un comando del paquete `VilCretas` que muestra este tipo de grafos de manera equivalente. Se puede visualizar su funcionamiento en:



`RegularGraph`: construye de forma pseudoaleatoria si es posible, un **grafo regular**. El comando requiere del paquete `Combinatorica`. Un grafo regular de orden k es aquel

donde cada vértice tiene el **mismo grado o valencia** y este valor corresponde a k . Las siguientes instrucciones generan un grafo regular de orden 3 con 6 vértices:

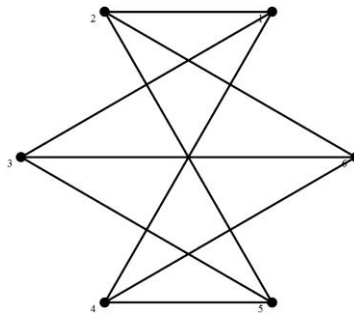
```
In[ ] :=
```

```
<< Combinatorica`
```

```
ShowGraph[RegularGraph[3, 6], VertexNumber -> True, PlotRange -> 0.1]
```

```
Quit[]
```

```
Out[ ] =
```



Al ser la salida pseudoaleatoria, el grafo variará de una ejecución a otra. `RegularQ` es otro comando de *Mathematica*, se usa para determinar si un grafo es regular. Por ejemplo:

```
In[ ] :=
```

```
<< Combinatorica`
```

```
ShowGraph[G = CompleteKPartiteGraph[2, 4], VertexNumber -> True, PlotRange -> 0.1];
```

```
RegularQ[G]
```

```
Quit[]
```

```
Out[ ] =
```

```
False
```

✳ La sentencia `GrafoRegular` del paquete `VilCretas` permite, además, generar este tipo de grafos. Se puede observar el siguiente video para aprender su empleo:



DodecahedralGraph: llamado **grafo dodecaedro**. El comando utiliza el paquete Combinatorica. Este grafo es estático pues no depende de ningún parámetro. Posee la apariencia:

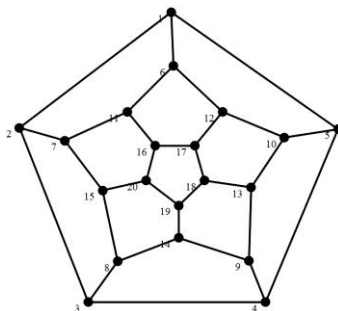
```
In[ ] :=
```

```
<< Combinatorica`
```

```
ShowGraph[DodecahedralGraph, VertexNumber -> True, PlotRange -> 0.1]
```

```
Quit[]
```

```
Out[ ] =
```



Este tipo de grafo puede ser representado, adicionalmente, usando la instrucción del paquete VilCretas denominada GrafoDato. Utilice esta sentencia, ¿qué diferencias se visualizan? Se puede consultar el video:






Como se recordará, toda relación tiene una forma de **representación matricial**, los grafos no son la excepción. La sección que sigue trata este importante tema.



Representaciones para un grafo

En el tema de relaciones binarias se introdujo un **arreglo bidimensional** que representa una relación R . Si particularmente R es un grafo G , a su matriz de representación se le llama **matriz de adyacencia** de G y se le denota como M_G . *Mathematica* brinda dos opciones para encontrar la matriz de adyacencia de un grafo, los comandos: `AdjacencyMatrix` y `ToAdjacencyMatrix`. Su diferencia radica en el uso del paquete `Combinatorica` por parte de `ToAdjacencyMatrix`.

Ejemplo 6

 Halle la matriz de adyacencia del grafo dodecaedro.

Al utilizar el software:

```
In[ ] :=
```

```
<< Combinatorica`
```

```
ShowGraph[G = DodecahedralGraph, VertexNumber -> True, PlotRange -> 0.1];
```

```
MatrixForm[ToAdjacencyMatrix[G]]
```

```
Quit[]
```

```
Out[ ] =
```


```

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

```



Ejemplo 7

 Determine la matriz de adyacencia de 10 grafos simples no dirigidos pseudoaleatorios con 5 vértices y 4 lados ¿Cuáles características tienen en común las matrices?

En *Mathematica* tenemos:

`In[] :=`

```

LG = RandomGraph[UniformGraphDistribution[5, 4], 10, VertexLabels -> "Name",
ImagePadding -> 10];

```

```

For[i = 1, i <= Length[LG], G = LG[[i]]; T = MatrixForm[AdjacencyMatrix[G]]; Print[G]; Print["La
matriz de adyacencia del grafo anterior es: ", T]; i++]

```

(10)



Al analizar cada una de las salidas obtenidas (no mostradas en este texto por su extensión), se observa que las matrices de adyacencia son **simétricas**, es decir, $(M_G)^t = M_G$. Una justificación de esta propiedad obedece a que toda relación simétrica implica una matriz de representación también simétrica. Si R es la relación de un grafo no dirigido, es natural que esta sea una relación simétrica dado que, si el nodo a está relacionado con el nodo b , existe una arista entre a y b , por consiguiente, al ser la arista no dirigida, b también está relacionado con a .

Otra singularidad observable, ocurre en la **diagonal principal** de cada matriz. Recordemos que las entradas con posiciones (i,i) (la fila es igual a la columna) conforman esta diagonal. En todas las matrices generadas, su diagonal principal es **nula**, es decir, los valores de las entradas son iguales a cero. Esto ocurre pues los grafos no contienen **ningún lazo**. En una matriz de adyacencia un uno en la diagonal principal, indica la existencia de un lazo en alguno de los vértices del grafo.



¡Estas dos propiedades son características comunes, en grafos simples no dirigidos!



En los ejemplos 6 y 7 se destaca el uso anidado de la instrucción `MatrixForm` con los comandos `AdjacencyMatrix` y `ToAdjacencyMatrix`. Lo anterior es recomendable con la finalidad de mostrar las matrices de adyacencia como un **arreglo rectangular** y no mediante otro tipo de estructura.



Un grafo $G = (V, E)$ posee **varias** matrices de adyacencia, dependiendo del **orden** de los elementos del conjunto V . Todas las matrices resultantes, eso sí, representan el mismo grafo. Para efectos del presente texto, el orden de los elementos de V se tomará por razones de simplicidad, en correspondencia con el orden del conjunto de los números reales o del abecedario.

Ejemplo 8



Considere el mismo enunciado del ejemplo 7 tomando los grafos pseudoaleatorios, dirigidos.

La solución propuesta usa el código 10 añadiendo dentro del RandomGraph el atributo DirectedEdges -> True.

! El alumno comprobará al **visualizar** las respuestas obtenidas que M_G **no necesariamente** es simétrica. Además, la diagonal principal de manera equivalente al ejemplo 7, está formada por **ceros**.

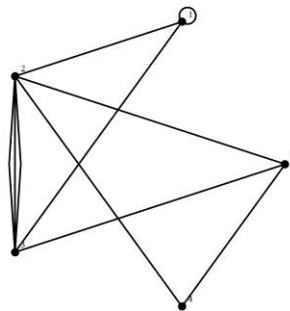


Si un grafo G incluye **lados múltiples**, la matriz de adyacencia M_G **deja de ser** una matriz booleana (entradas con unos y ceros). Si en los vértices v_i y v_j ocurren k aristas múltiples, en la entrada (i, j) de M_G , se escribe el valor de k .

Ejemplo 9



Encuentre la matriz de adyacencia del grafo:



(11)

En Mathematica:

```
In[ ] :=
```

```
<< Combinatorica`
```


```

h = {{1, 1}, {1, 2}, {1, 3}, {2, 3}, {2, 3}, {2, 3}, {2, 4}, {4, 5}, {5, 3}, {5, 2}};
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black,
EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1]
MatrixForm[ToAdjacencyMatrix[G]]
Quit[]


```

Out[] =

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 3 & 1 & 1 \\ 1 & 3 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

 La matriz M_G de este ejemplo **no es booleana** (no todas las entradas son ceros o unos) pues el grafo G tiene lados múltiples. El 3 en las posiciones (2,3) y (3,2) de la matriz de adyacencia, simboliza la existencia de tres lados múltiples entre los vértices 2 y 3.




 En el paquete VilCretas se encuentra disponible la instrucción MGrafo para construir la matriz de adyacencia de un grafo cualesquiera. Se sugiere su exploración observando:



Como ya hemos descrito, dado un grafo, este tiene asociado una matriz de adyacencia M_G que lo representa. Recíprocamente, dada la matriz M_G se puede construir el grafo G . En esta dirección, *Mathematica* facilita los comandos `AdjacencyGraph` y `FromAdjacencyMatrix`. Ambos reciben como parámetros la matriz de adyacencia M_G y

gestan como salida a G . FromAdjacencyMatrix necesita del paquete Combinatorica.

Ejemplo 10

 Dadas las siguientes matrices de adyacencia represente con Mathematica cada grafo.

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

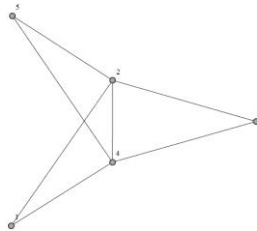
Por medio del comando AdjacencyGraph se produce:

$In[] :=$

$M_G = \{\{0, 1, 0, 1, 0\}, \{1, 0, 1, 1, 1\}, \{0, 1, 0, 1, 0\}, \{1, 1, 1, 0, 1\}, \{0, 1, 0, 1, 0\}\};$

AdjacencyGraph[M_G, VertexLabels -> "Name", ImagePadding -> 10]

Out[] =

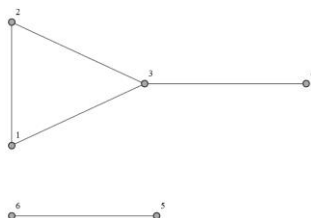


$In[] :=$

$M_G = \{\{0, 1, 1, 0, 0, 0\}, \{1, 0, 1, 0, 0, 0\}, \{1, 1, 0, 1, 0, 0\}, \{0, 0, 1, 0, 0, 0\}, \{0, 0, 0, 0, 0, 1\}, \{0, 0, 0, 0, 1, 0\}\};$

AdjacencyGraph[M_G, VertexLabels -> "Name", ImagePadding -> 10]

Out[] =



Con FromAdjacencyMatrix:

In[] :=

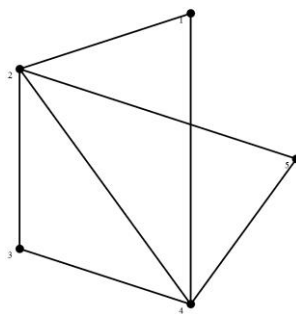
<< Combinatorica`

$M_G = \{\{0, 1, 0, 1, 0\}, \{1, 0, 1, 1, 1\}, \{0, 1, 0, 1, 0\}, \{1, 1, 1, 0, 1\}, \{0, 1, 0, 1, 0\}\};$

ShowGraph[FromAdjacencyMatrix[M_G], VertexNumber -> True, PlotRange -> 0.1]

Quit[]

Out[] =



In[] :=

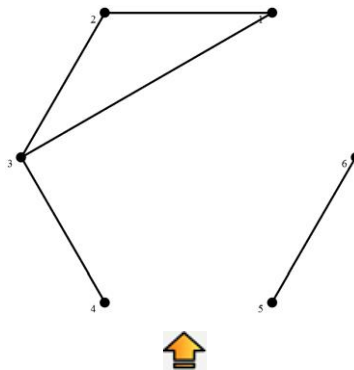
<< Combinatorica`

$M_G = \{\{0, 1, 1, 0, 0, 0\}, \{1, 0, 1, 0, 0, 0\}, \{1, 1, 0, 1, 0, 0\}, \{0, 0, 1, 0, 0, 0\}, \{0, 0, 0, 0, 0, 1\}, \{0, 0, 0, 0, 1, 0\}\};$

ShowGraph[FromAdjacencyMatrix[M_G], VertexNumber -> True, PlotRange -> 0.1]


Quit[]

Out[] =



La matriz de adyacencia de un grafo simple tiene una **notable aplicación**. Consideremos el teorema próximo.

Teorema 3

 Sea $G=(V,E)$ un grafo simple, $V=\{v_1,v_2,\dots,v_n\}$ y M_G la matriz de adyacencia de G en el orden del conjunto V . $(M_G)^k=(c_{ij})$, con $k \in \mathbb{N}$ es una matriz donde la entrada c_{ij} representa el número de rutas (con repetición de aristas) de longitud k , del nodo v_i al nodo v_j en G .



MatrixPower en *Mathematica* calcula la potencia k -ésima de una matriz.

Ejemplo 11



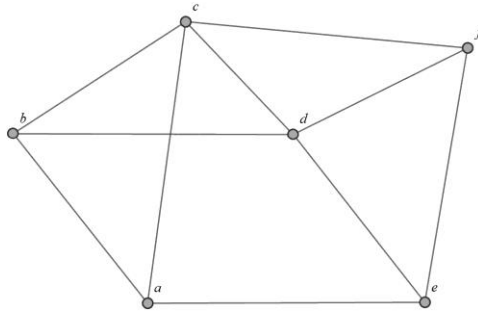
Desarrolle con ayuda de software, una función que determine el número de rutas de longitud k entre dos vértices a y b , sobre un grafo simple G . Suponga que se permite la repetición de aristas en las trayectorias.

El teorema 3 da cuenta sobre el procedimiento empleado en la resolución de este ejercicio. Se asume como parámetros de la función: el grafo, el valor de k y, los nodos a y b . Luego:

```
CantRutasLk[G_Graph,k_,a_,b_] := Module[{}, L = VertexList[G]; If[MemberQ[L, a] == True && MemberQ[L, b] == True, T = AdjacencyMatrix[G]; P = MatrixPower[T, k]; For[i = 1, i <= VertexCount[G], If[a == L[[i]], posa = i]; If[b == L[[i]], posb = i]; i++]; Print["La cantidad de rutas de longitud ", k, " con repeticiones en las aristas del nodo ", a, " al nodo ", b, " es: ", P[[{posa, posb}]], Print["Alguno de los nodos no forma parte del grafo"]]]
```

(12)

Por ejemplo, si se corre esta función en el grafo con respecto a los vértices a y f , y $k=3$:



(13)

Devuelve:

$Out[] =$

La cantidad de rutas de longitud 3 con repeticiones en las aristas del nodo a al nodo f es: 4

Las cuatro rutas por inferencia en G son: $a \rightarrow b \rightarrow c \rightarrow f$, $a \rightarrow b \rightarrow d \rightarrow f$, $a \rightarrow c \rightarrow d \rightarrow f$ y $a \rightarrow e \rightarrow d \rightarrow f$.



Una sentencia del paquete VilCretas que pudo haber resuelto lo anterior, se llama CantMRutas. Se recomienda al estudiante consultar:



Los grafos gozan de **otro tipo** de representación bidimensional, llamada **matriz de incidencia**.

Definición 4



Sea $G=(V,E)$ un grafo no dirigido con $V=\{v_1,v_2,\dots,v_n\}$ y $E=\{e_1,e_2,\dots,e_m\}$. Se llama matriz de incidencia de G en el orden de los elementos de V y E , a la matriz denotada M'_G , $M'_G=(m_{ij})$ de tamaño n por m , tal que:

$$m_{ij} = \begin{cases} 1 & \text{si la arista } e_j \text{ es incidente sobre el v\u00e9rtice } v_i \\ 0 & \text{en caso contrario} \end{cases}$$



En un grafo simple G la **suma** de las entradas de la fila i -ésima de la matriz de incidencia, da como resultado la **valencia o grado** del v\u00e9rtice v_i . Si el grafo tiene lados m\u00faltiples esta propiedad se contin\u00faa satisfaciendo. Por otra parte, si G posee lazos la propiedad anterior **no se cumple**.

La definici\u00f3n 4 es **generalizable** a un digrafo, sin embargo, dada su poca aplicabilidad pr\u00e1ctica, se omitir\u00e1 dicho concepto de este texto.




En el *software Mathematica* yace la instrucci\u00f3n `IncidenceMatrix`. A trav\u00e9s de ella, se encuentra la matriz de incidencia de un grafo creado con `Graph` o `ShowGraph`. Si el grafo G se ha construido con `Graph` y tiene lazos, es imprescindible se\u00f1alar, que `IncidenceMatrix[G]` **no retorna** una matriz booleana. La entrada (i,j) siendo v_i el nodo donde ocurre el ciclo en la arista e_j , se toma como un **dos**, en lugar de un uno. Esto contradice la definici\u00f3n 4. Se sugiere al estudiante evitar esta ambig\u00fcedad, utilizando siempre la sentencia `ShowGraph` si el grafo contiene **alg\u00fan lazo** y se requiere su matriz de incidencia.



Cuando un grafo G tiene lados m\u00faltiples por cada arista m\u00faltiple, se toma una columna m\u00e1s en la matriz de incidencia de G .

Ejemplo 12

 Halle la matriz de incidencia del grafo 13.

En *Mathematica*:

In[] :=

```
G = Graph[{a ↔ b, a ↔ c, b ↔ c, c ↔ d, d ↔ b, e ↔ a, e ↔ f, f ↔ d, f ↔ c, e ↔ d},
```

```
VertexLabels -> "Name", ImagePadding -> 10];
```

```
MatrixForm[IncidenceMatrix[G]]
```

Out[] =

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

El lector al sumar los elementos de cada fila i -ésima, puede comprobar su correspondencia con el grado del vértice v_i .

Otra forma de **visualización** de la matriz de incidencia reside en construir una tabla donde las filas tengan como etiquetas los nodos del grafo y las columnas, sus aristas. En el *software*, esto se efectúa así:

In[] :=

```
G = Graph[{a ↔ b, a ↔ c, b ↔ c, c ↔ d, d ↔ b, e ↔ a, e ↔ f, f ↔ d, f ↔ c, e ↔ d},
```

```
VertexLabels -> "Name", ImagePadding -> 10];
```

```
TableForm[Normal@IncidenceMatrix[G], TableHeadings -> {Style[#, Black] & /@  
VertexList[G], Style[#, Black] & /@ EdgeList[G]}]
```

Out[] =

	a ↔ b	a ↔ c	b ↔ c	c ↔ d	d ↔ b	e ↔ a	e ↔ f
a	1	1	0	0	0	1	0
b	1	0	1	0	1	0	0
c	0	1	1	1	0	0	0
d	0	0	0	1	1	0	0
e	0	0	0	0	0	1	1
f	0	0	0	0	0	0	1

El código funciona siempre y cuando, el grafo se haya construido mediante Graph.



Ejemplo 13



Determine la matriz de incidencia del grafo 11.

En el software:

`In[] :=`

`<< Combinatorica``

`h = {{1, 1}, {1, 2}, {1, 3}, {2, 3}, {2, 3}, {2, 3}, {2, 4}, {4, 5}, {5, 3}, {5, 2}};`

`ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black, EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1];`

`MatrixForm[IncidenceMatrix[G]]`

`Quit[]`

`Out[] =`

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Si se suman las entradas de la primera fila de la matriz de incidencia anterior, el resultado es 3. Este valor **no es igual** a la valencia del nodo 1, pues en 1 ocurre un lazo.



En los ejemplos 12 y 13, se observa la anidación de `MatrixForm` con `IncidenceMatrix`. Esto es similar a lo comentado en la página 42. Se sugiere al lector

siempre utilizar la instrucción `MatrixForm`, para visualizar la matriz de incidencia como un **arreglo rectangular**.

Análogamente a lo establecido en la página 44, dada la matriz de incidencia de un grafo G , es posible generar a G . *Mathematica* nos dota con el comando `IncidenceGraph`, que efectúa este **procedimiento inverso** sobre un grafo simple.

Ejemplo 14



Represente con *Mathematica* el grafo dado por la matriz de incidencia:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

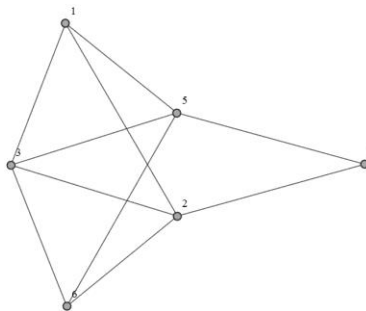
En el *software*:

`In[] :=`

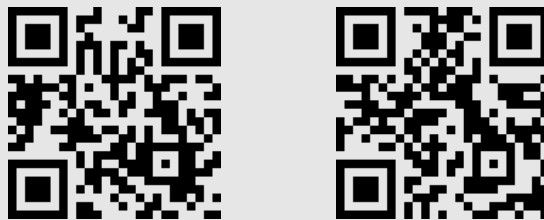
`W = {{1, 1, 0, 0, 0, 0, 0, 1, 0, 0}, {1, 0, 1, 1, 0, 1, 0, 0, 0, 0}, {0, 1, 1, 0, 1, 0, 0, 0, 0, 1}, {0, 0, 0, 1, 0, 0, 0, 0, 1, 0}, {0, 0, 0, 0, 1, 0, 1, 1, 1, 0}, {0, 0, 0, 0, 0, 1, 1, 0, 0, 1}};`

`IncidenceGraph[W, VertexLabels -> "Name", ImagePadding -> 10]`

`Out[] =`



✦ En la página 19 se comentó la funcionalidad de la sentencia `WeightedAdjacencyMatrix`, que retorna la **matriz de adyacencia de pesos** en un grafo ponderado. *Mathematica* cuenta también, con el comando `WeightedAdjacencyGraph`. Este recibe la matriz de adyacencia de pesos de un grafo simple y lo devuelve dibujado en el plano. Omitiremos la realización de ejemplos a este respecto, quedando esta tarea como un quehacer futuro para el lector, en la parte de ejercicios de la presente unidad didáctica. Se sugiere al alumno, además, consultar los comandos del paquete *VilCretas*: `MIGrafo` y `MPGrafo`, que resuelven adicionalmente, la construcción de las matrices de incidencia y de pesos, respectivamente. Para ello, se pueden visualizar los videos:




La sección que continúa trata el estudio de los **circuitos sobre un grafo**. Los circuitos constituyen una unidad de conocimiento esencial por sus distintas aplicaciones en informática u otros campos. De allí que se dedique un apartado exclusivo para desarrollar el tema.



Circuitos en un grafo

Iniciaremos con el **concepto de circuito**.

Definición 5

 Sea $G = (V, E)$ un grafo. A cualquier ruta sobre G que inicie y finalice en el mismo vértice se le llama **circuito**. Si un circuito no repite nodos se dice que es **simple**. Si un circuito contiene todas las aristas de G se denomina **circuito de Euler**. Si un circuito contiene todos los vértices de G sin repetición de nodos, se llama **circuito de Hamilton**. Una ruta que incluye todos los lados de G sin ser un circuito, se define como un **camino de Euler** o **trayectoria euleriana**. Cualquier trayectoria que no sea un circuito y que contenga una única vez todos los vértices de G , se llama **ruta de Hamilton** o **camino hamiltoniano**.

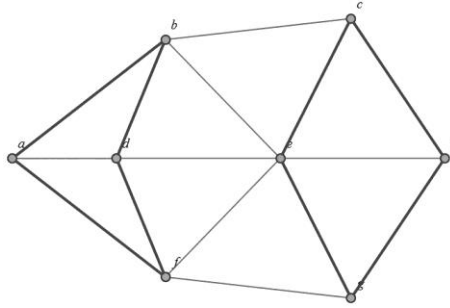
Un **lazo** sobre un grafo que lo contenga es en sí mismo, un **circuito** de acuerdo con la definición 5. El código presentado a continuación, resalta dos circuitos distintos sobre un mismo grafo:

```
In[ ] :=
```

```
Graph[{a ↔ b, a ↔ d, a ↔ f, b ↔ c, b ↔ d, b ↔ e, c ↔ e, c ↔ z, d ↔ e, d ↔ f, e ↔ g, e ↔ f, f ↔ g, g ↔ z, z ↔ e}, VertexLabels -> "Name", ImagePadding -> 10, GraphHighlight -> {a ↔
```

b, b—d, d—f, f—a, c—z, z—g, g—e, e—c}, GraphHighlightStyle -> "Thick"]

Out[] =



✶ En el *software Mathematica* existe el comando `FindCycle` del paquete *Combinatorica*. Este **determina un circuito** sobre un grafo G . Si G no tiene circuitos retorna el conjunto vacío. También, se cuenta con la instrucción `Girth` de *Combinatorica*, encargada de devolver la **longitud del circuito más pequeño** sobre G . Si G no presenta circuitos, `Girth` por defecto es igual a ∞ . Por ejemplo:

In[] :=

<< *Combinatorica`*

h = {{1, 2}, {1, 3}, {2, 4}, {2, 6}, {3, 4}, {3, 5}, {4, 5}, {4, 6}};

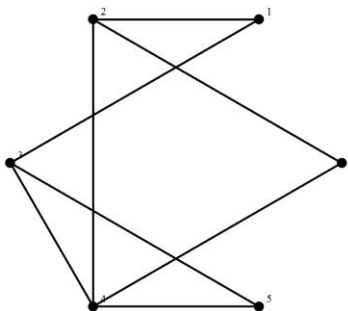
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black, EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1]

FindCycle[G]

Girth[G]

Quit[]

Out[] =



{3, 1, 2, 4, 3}

3

El resultado de FindCycle se puede desplegar gráficamente en *Mathematica*. En este caso, se procede así:

```
In[ ] :=
```

```
<<Combinatorica`
```

```
h = {{1, 2}, {1, 3}, {2, 4}, {2, 6}, {3, 4}, {3, 5}, {4, 5}, {4, 6}};
```

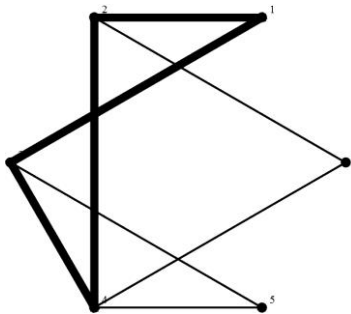
```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black,  
EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1];
```

```
v = FindCycle[G];
```

```
ShowGraph[Highlight[G, {Partition[v, 2, 1]}], VertexLabel -> True, PlotRange -> 0.1]
```

```
Quit[]
```

```
Out[ ] =
```



`AcyclicGraphQ[G]` es otra función booleana del *software*, con valor lógico True si G es un grafo simple sin circuitos. A un grafo con esta propiedad se le llama: **acíclico**.



En el paquete *VilCretas* se cuenta con la instrucción `Circuito` que retorna un circuito en un grafo dado. El uso del comando se puede consultar en:

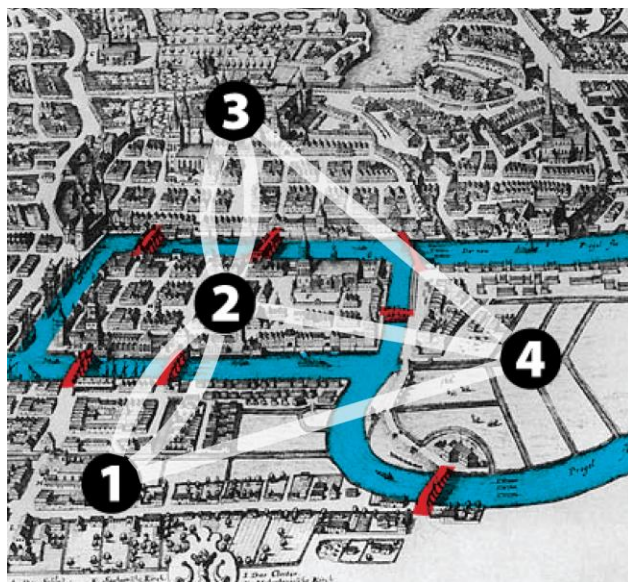


En la historia de las matemáticas se considera a *Leonhard Euler* el **padre** de la teoría de grafos. La razón de ello obedece a una publicación que *Euler* realizó en 1736, donde da



respuesta a un famoso problema conocido como: el **paso de los siete puentes de Königsberg** (Castro, 1988). Siendo *Leonhard Euler* un **genio** de su época, se le planteó la siguiente interrogante: por la ciudad de *Königsberg* (en la actualidad *Kaliningrado, Rusia*) pasa el río *Pregel*. Sobre él, se encuentran dos islas comunicadas entre sí y con las riveras, por siete puentes ¿Es posible realizar un recorrido circuital que pase una única vez por cada puente? Esta situación se representa con

mayor claridad en la siguiente figura:



Lo anterior forma un grafo cuyos **nodos** son las **islas y las riveras**, mientras que sus **aristas** están constituidas por los **siete puentes** de *Königsberg*. En *Mathematica*:

```
In[ ] :=
```

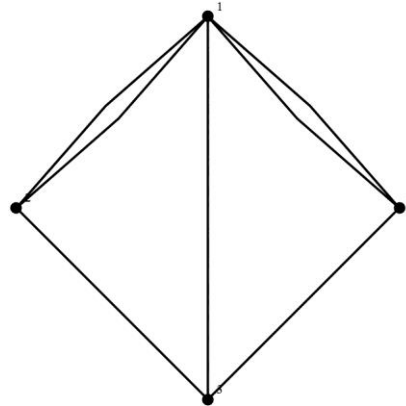
```
<< Combinatorica`
```

```
h = {{1, 2}, {1, 2}, {1, 4}, {1, 4}, {1, 3}, {2, 3}, {3, 4}};
```

```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black,
EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1]
```

```
Quit[]
```

$Out[] =$



(15)


Los vértices 1 y 3 del grafo (15), representan las islas y, los nodos 2 y 4 las riveras. *Euler demostró* que un recorrido circuital con todos los lados del grafo (15) es **imposible**. El razonamiento empleado se basó en observar las **valencias** de cada uno de los vértices de (15). Si un nodo no es un punto de partida, necesariamente si se entra a él, se debe contar con una arista distinta que permita salir. Esto significa que cada vértice debe tener una **valencia par**. En (15) todos los grados son impares, por este motivo, es imposible formar un circuito de *Euler*.

✦ Un grafo que representa el problema de los puentes de *Königsberg* puede ser generado a través de la instrucción `GrafoKönigsberg` del paquete `VilCretas`:



! Lo mencionado con respecto a la solución del problema de los siete puentes de *Königsberg*, constituye un **teorema de caracterización** sobre la **existencia** de un circuito de *Euler* en un grafo conexo.

Teorema 4

 Sea $G = (V, E)$ un grafo conexo. Si todas las valencias de los vértices de G son pares, G tiene un circuito de Euler. Además, si hay al menos un grado impar en los nodos de G , el grafo no posee un circuito de Euler.

Ejemplo 15



Elabore un programa en Mathematica que determine si un grafo sin ciclos posee un circuito de Euler.

Utilizando el software y el teorema 4, se tiene:

```
<< Combinatorica`
```

```
BooleanCircuitoEuler[G_Graph] := Module[{v = Degrees[G], j = 0}, If[EdgeConnectivity[G] != 0, For[i = 1, i <= Length[v], If[OddQ[v[[i]]] == True, Print["False"]; j = 1; Break[]; i++]; If[j == 0, Print["True"], Print["False"]]]
```

Por ejemplo:

```
In[ ] :=
```

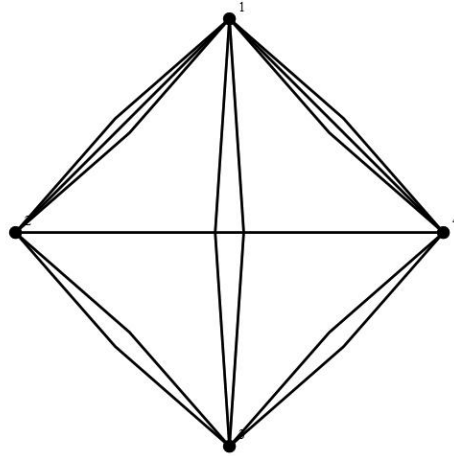
```
h = {{1, 2}, {1, 2}, {1, 2}, {1, 4}, {1, 4}, {1, 4}, {1, 3}, {2, 3}, {2, 3}, {2, 4}, {3, 4}, {3, 4}, {1, 3}};
```

```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black, EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1]
```

```
BooleanCircuitoEuler[G]
```

```
Quit[]
```

```
Out[ ] =
```



True

! Más directamente, *Mathematica* presenta los comandos: `EulerianQ` y `EulerianGraphQ`. Ambos, devuelven `True` si el argumento de la instrucción es un grafo G sin lazos, con circuitos de *Euler*. El alumno debe prestar atención al utilizar `EulerianQ` sobre un grafo G con ciclos. `EulerianQ[G]` retorna `False` sobre grafos con lazos que sí tienen circuitos de *Euler*. Por este motivo, se recomienda el uso de `EulerianGraphQ` si el grafo G presenta ciclos en su estructura. También, se debe señalar que `EulerianQ` forma parte del paquete `Combinatorica`.



✦ Se sugiere al estudiante ante el problema de los circuitos eulerianos, consultar también, el comando `CircuitosEuler` del paquete `VilCretas`:



Ejemplo 16



Conjeture con ayuda de software, ¿qué condiciones debe cumplir un grafo completo K_n para tener circuitos de Euler? ¿Cuáles condiciones se deben satisfacer en un grafo bipartito completo $K_{n,m}$?

Analizaremos en primera instancia K_n . Se recurrirá a la sentencia EulerianGraphQ mediante un ciclo que genera distintos órdenes del grafo:

```
In[ ] :=
```

```
For[i = 3, i <= 12, G = CompleteGraph[i]; Print["Para el orden ", i, " el valor lógico sobre la existencia de un circuito de Euler es: ", EulerianGraphQ[G]]; i++]
```

```
Out[ ] =
```

Para el orden 3 el valor lógico sobre la existencia de un circuito de Euler es: True

Para el orden 4 el valor lógico sobre la existencia de un circuito de Euler es: False

Para el orden 5 el valor lógico sobre la existencia de un circuito de Euler es: True

:

La salida muestra 10 casos hasta el orden 12 de K_n . Los resultados permiten conjeturar:

“ K_n posee circuitos de Euler sí y solo sí n es impar”.

Con respecto a $K_{n,m}$ se consideran también, distintos valores particulares. Se usa la instrucción EulerianQ en lugar de EulerianGraphQ, pues CompleteKPartiteGraph que construye los grafos bipartitos completos en cada iteración, requiere del paquete Combinatorica:

```
In[ ] :=
```

```
<< Combinatorica`
```

```
For[i = 1, i <= 6, For[j = 1, j <= 6, G = CompleteKPartiteGraph[i, j]; Print["Para el orden ", i, "x", j, " el valor lógico sobre la existencia de un circuito de Euler es: ", EulerianQ[G]]; j++]; i++]
```

```
Quit[]
```

```
Out[ ] =
```


Para el orden 1x1 el valor lógico sobre la existencia de un circuito de Euler es: False
Para el orden 1x2 el valor lógico sobre la existencia de un circuito de Euler es: False
Para el orden 1x3 el valor lógico sobre la existencia de un circuito de Euler es: False
⋮

De las 36 observaciones sobre los correspondientes grafos bipartitos completos, se deduce que: “ $K_{n,m}$ tiene circuitos de Euler sí y solo sí n y m son números pares”.



Otra situación importante, radica en conocer cuál es un circuito de *Euler* sobre un grafo G que lo contiene. Para ello, es familiar en teoría de grafos un método denominado: **algoritmo de Fleury**.

Teorema 5

 Sea $G=(V,E)$ un grafo conexo con grados pares en todos sus vértices. Un circuito de Euler sobre G se forma así:

- Se parte de un vértice v cualquiera de G y se agrega a L una arista inicial e con extremo v , que no sea un puente (un puente es un lado del grafo, que al ser eliminado, lo transforma en desconexo). Sea el conjunto de aristas seleccionables: CAS , $CAS = E - \{e\}$.
- Si $CAS = \phi$, se ha finalizado, L contiene el circuito de Euler.
- Se escoge un nuevo lado e_N para el circuito a construir, siempre y cuando e_N no sea un puente y sea adyacente a la última arista de L . Si la única arista elegible e_N es un puente, se selecciona.
- Sea $CAS = CAS - \{e_N\}$ y $L = L \cup \{e_N\}$, quedando e_N como último elemento de L . Vaya al paso dos.

Ejemplo 17



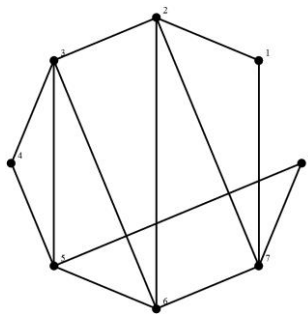
Desarrolle una implementación en Mathematica del algoritmo de Fleury sobre un grafo simple.

Usando Mathematica:

```
<< Combinatorica`
```

```
CircuitoEuleriano[G_Graph] := Module[{GL = G}, If[EulerianQ[G] == True, CAS = Edges[GL]; L = Flatten[CAS[[1]]]; GL = DeleteEdge[GL, CAS[[1]]]; CAS = Delete[CAS, Position[CAS, CAS[[1]]]]; While[Length[Edges[GL]] != 0, CAI = {}; For[i = 1, i <= Length[CAS], w = Last[L]; If[w == CAS[[i, 1]] || w == CAS[[i, 2]], CAI = Append[CAI, CAS[[i]]]; CAI = DeleteDuplicates[CAI]; i++]; For[i = 1, i <= Length[CAI], If[MemberQ[Bridges[GL], CAI[[i]]] == False, GL = DeleteEdge[GL, CAI[[i]]]; If[CAI[[i, 2]] != w, L = Append[L, CAI[[i, 2]]], L = Append[L, CAI[[i, 1]]]; CAS = Delete[CAS, Position[CAS, CAI[[i]]]; Break[], If[i == Length[CAI], GL = DeleteEdge[GL, CAI[[i]]]; If[CAI[[i, 2]] != w, L = Append[L, CAI[[i, 2]]], L = Append[L, CAI[[i, 1]]]; CAS = Delete[CAS, Position[CAS, CAI[[i]]]; i++]; Print["Un circuito de Euler mediante una representación de nodos es: ", L]; Print["El mismo circuito representado a través de aristas es: ", Partition[L, 2, 1]], Print["El grafo no tiene circuitos de Euler"]]]
```

Por ejemplo, al correr la función `CircuitoEuleriano` en:



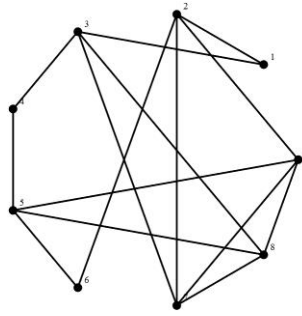
Se obtiene:

Un circuito de Euler mediante una representación de nodos es: {1, 2, 3, 4, 5, 3, 6, 2, 7, 6, 5, 8, 7, 1}

El mismo circuito representado a través de aristas es:

{1, 2}, {2, 3}, {3, 4}, {4, 5}, {5, 3}, {3, 6}, {6, 2}, {2, 7}, {7, 6}, {6, 5}, {5, 8}, {8, 7}, {7, 1}

Si se corre ahora CircuitoEuleriano, sobre:



Retorna:

Un circuito de Euler mediante una representación de nodos es: {1, 2, 6, 5, 4, 3, 7, 2, 9, 5, 8, 7, 9, 8, 3, 1}


El mismo circuito representado a través de aristas es: {{1, 2}, {2, 6}, {6, 5}, {5, 4}, {4, 3}, {3, 7}, {7, 2}, {2, 9}, {9, 5}, {5, 8}, {8, 7}, {7, 9}, {9, 8}, {8, 3}, {3, 1}}




Es importante indicar que en el paquete VilCretas se facilita la instrucción *Fleury* la cual muestra paso a paso la aplicación del algoritmo de *Fleury*. Para obtener más información se puede consultar:




Mathematica encuentra también circuitos de *Euler* sobre un grafo dado, al utilizar los comandos: *EulerianCycle* y *FindEulerianCycle*. La primera instrucción devuelve una lista de vértices que generan el circuito euleriano. La segunda, forma un vector de aristas que representan el circuito de *Euler*. *FindEulerianCycle*[*G*, *k*] halla el número máximo *k* de circuitos de *Euler* sobre *G*. Si el grafo no tiene un circuito euleriano, las dos sentencias retornan el conjunto vacío. *EulerianCycle* es parte del paquete *Combinatorica*.

 En ciertos grafos que no poseen un circuito de *Euler* es posible determinar al menos una **ruta** de *Euler*. Las trayectorias eulerianas como ya se mencionó en la definición 5, incluyen **todas las aristas** de un grafo donde el nodo inicial es **distinto** del vértice final. Consideremos el siguiente teorema.

Teorema 6

 Sea $G=(V,E)$ un grafo conexo. Si G tiene exactamente dos nodos a y b de valencia impar, G posee una ruta de Euler que une el vértice a con el nodo b . Si en G hay más o menos de dos vértices con grado impar entonces G no contiene ningún camino euleriano.

Ejemplo 18

 Elabore con Mathematica una función booleana que determine si dado un grafo sin ciclos, este contiene una trayectoria de Euler.

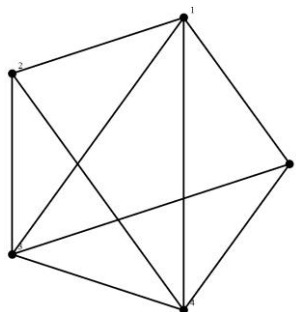
Recurriendo a ideas similares a las presentadas en el ejemplo 15, tenemos:

<< Combinatorica`

```
BooleanRutaEuler[G_Graph] := Module[{v = Degrees[G], j = 0}, If[EdgeConnectivity[G] != 0,
For[i = 1, i <= Length[v], If[OddQ[v[[i]]] == True, j++; i++]; If[j == 2, Print["True"], Print["False"]],
Print["False"]]]
```

(16)

Al correr el código (16), en el grafo:



Se devuelve el valor lógico: True.





El algoritmo de *Fleury* es **modificable** con la intención de encontrar un camino de *Euler* en un grafo que satisfaga las condiciones del teorema 6 ¿Cómo se podría realizar lo anterior? Esta tarea quedará como un ejercicio para el lector.

Los circuitos de *Hamilton* constituyen otro tema esencial. No existe **ningún teorema que caracterice** la existencia de un circuito de *Hamilton* sobre un grafo conexo, como sí ocurre con los circuitos de *Euler*, sin embargo, por medio de algunas propiedades del grafo en estudio, es posible determinar la **existencia** de un circuito de *Hamilton*. Veamos el teorema que se expone a continuación.

Teorema 7



Sea $G = (V, E)$ un grafo simple y conexo, con n nodos y m lados, $n > 2$, luego:

- (Propiedad de Ore) Si para todo par de vértices no adyacentes a y b de V , se cumple: $\theta(a) + \theta(b) \geq n$ entonces G tiene un circuito de Hamilton.
- (Propiedad de Dirac) Si para todo vértice a de V , $\theta(a) \geq \frac{n}{2}$ entonces G posee un circuito hamiltoniano.
- Si $m \geq \frac{n^2 - 3n + 6}{2}$ entonces G contiene un circuito de Hamilton.

Ninguna de las propiedades enunciadas en el teorema 7 son de **caracterización**, es decir, si un grafo G no cumple alguna de ellas, **no es posible** concluir que G no contiene un circuito de *Hamilton*. Inclusive si G no satisface ninguna de las propiedades, la única respuesta válida, consiste en afirmar la **ausencia de un criterio** para determinar la existencia de un circuito hamiltoniano.

Ejemplo 19



Elabore una rutina en *Mathematica* por medio del teorema 7, donde al recibir un grafo indique si éste tiene un circuito de *Hamilton*, o bien, no hay criterio.

En el software:

```
ExisteCircuitoHamilton[G_Graph] := If[SimpleGraphQ[G] == True && ConnectedGraphQ[G]
== True && VertexCount[G] > 2, Module[{Vertices = VertexList[G], v = VertexDegree[G], n =
```

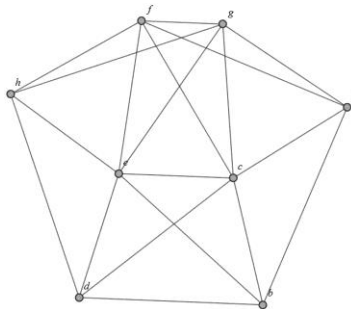
```

VertexCount[G], m = EdgeCount[G], prop1 = 1, h = 1, prop2 = 1, prop3 = 1}, For[i = 1, i <= n, For[j =
i, j <= n, If[i != j && EdgeQ[G, Vertices[[i]] ↔ Vertices[[j]]] == False, h = 0; If[VertexDegree[G,
Vertices[[i]]] + VertexDegree[G, Vertices[[j]]] < n, prop1 = 0; Break[ ]; Break[ ]]; j++]; i++]; For[i
= 1, i <= Length[v], If[v[[i]] < n/2, prop2 = 0; Break[ ]; i++]; If[m < (n^2 - 3 n + 6)/2, prop3 = 0];
If[prop1 == 1 && h == 1, Print["No se puede verificar la propiedad de Ore pues todos los
vértices son adyacentes entre sí"]; If[prop1 == 1 && h == 0, Print["Se satisface la propiedad
de Ore"]; If[prop1 == 0, Print["No se satisface la propiedad de Ore"]; If[prop2 == 0, Print["No
se cumple la propiedad de Dirac"], Print["Se cumple la propiedad de Dirac"]; If[prop3 == 0,
Print["No se satisface la propiedad 3"], Print["Se satisface la propiedad 3"]; If[prop1 == 1 ||
prop2 == 1 || prop3 == 1, Print["Existe un circuito de Hamilton"], Print["No hay criterio"]],
Print["El grafo no es simple, no es conexo o hay una cantidad insuficiente de vértices"]]

```



Por ejemplo, ExisteCircuitoHamilton corrido sobre:



(17)

Da la respuesta:

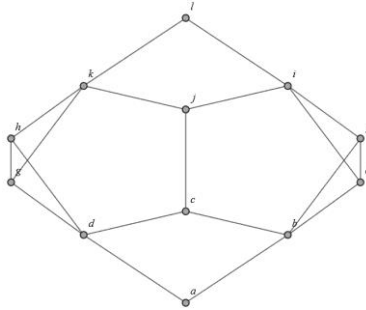
Se satisface la propiedad de Ore

Se cumple la propiedad de Dirac

No se satisface la propiedad 3

Existe un circuito de Hamilton

Utilizando la función ExisteCircuitoHamilton, en:



Devuelve:

No se satisface la propiedad de Ore

No se cumple la propiedad de Dirac

No se satisface la propiedad 3

No hay criterio

Las instrucciones `HamiltonianQ` y `HamiltonianGraphQ` de *Mathematica*, tienen la función de determinar si un **grafo posee** un circuito hamiltoniano. `HamiltonianQ` se diferencia de `HamiltonianGraphQ`, pues la primera sentencia usa el paquete `Combinatorica`. Por ejemplo:

```
In[ ] :=
```

```
<< Combinatorica`
```

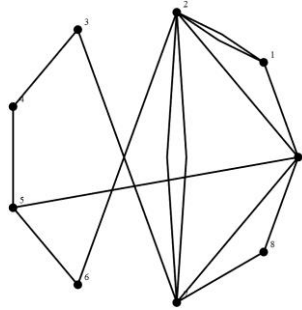
```
h = {{1, 2}, {1, 2}, {1, 9}, {2, 6}, {2, 7}, {2, 7}, {2, 9}, {3, 4}, {3, 7}, {4, 5}, {5, 6}, {5, 9}, {7, 8}, {7, 9}, {8, 9}};
```

```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black, EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1]
```

```
HamiltonianQ[G]
```

```
Quit[]
```

```
Out[ ] =
```



True

☀ Se sugiere al estudiante consultar también, los comandos `CircuitoHamiltonQ` y `CircuitosHamilton` del paquete `VilCretas`, los cuales, de una manera muy interesante, resuelven el problema de los circuitos hamiltonianos:



Ejemplo 20

🌀 Conjeture cuál condición debe cumplir $K_{n,m}$ para contener un circuito de Hamilton.

En *Mathematica*, al analizar 25 casos, se tiene:

`In[] :=`

`<< Combinatorica``

`For[i = 1, i <= 5, For[j = 1, j <= 5, G = CompleteKPartiteGraph[i, j]; Print["Para el orden ", i, "x", j, " el valor lógico sobre la existencia de un circuito de Hamilton es: ", HamiltonianQ[G]]; j++];`

`i++]`

`Quit[]`


`Out[] =`

Para el orden 1x1 el valor lógico sobre la existencia de un circuito de Hamilton es: False

⋮

Observando las respuestas generadas por el *software*, se conjetura: “ $K_{n,m}$ tiene un **circuito de Hamilton** sí y solo sí $n = m$, $n \geq 2$ ”.



 Si un grafo G contiene un circuito de *Hamilton*, se retorna en *Mathematica* a través de los comandos: `HamiltonianCycle` o `FindHamiltonianCycle`. `HamiltonianCycle` es una instrucción del paquete `Combinatorica` y acepta a diferencia de `FindHamiltonianCycle`, el atributo `All`. `All` solicita a *Mathematica* todos los circuitos de *Hamilton* sobre el grafo G .

Ya se comprobó en el grafo (17) la existencia de circuitos de *Hamilton*. `FindHamiltonianCycle[G]` devuelve uno de ellos:

`Out[] =`

`{{b → a, a → f, f → h, h → d, d → e, e → g, g → c, c → b}}`



También, en algunos grafos donde no es posible hallar un circuito hamiltoniano, existe al menos una trayectoria o camino de *Hamilton*. Una ruta de *Hamilton* según la definición 5, es una trayectoria que contiene una única vez todos los nodos del grafo. `HamiltonianPath[G]` del paquete `Combinatorica` determina si existe, una trayectoria hamiltoniana. Si se añade `All` dentro de la sentencia, retorna todos los caminos de *Hamilton* posibles sobre G .



Un problema relacionado con circuitos de *Hamilton* recibe el nombre de: **problema del agente viajero**. Este consiste en encontrar, en un grafo con pesos, un circuito de *Hamilton* de longitud mínima. Se denomina agente viajero pues resuelve la situación, donde se asume que los vértices del grafo representan los lugares que un viajero debe visitar y los pesos de los lados, **costos** asociados a cada viaje. Luego, el agente viajero está interesado en conocer la trayectoria que le permita incurrir en **gastos mínimos**, visitando todos los sitios hasta llegar nuevamente al punto de partida. *Mathematica* integra el comando `TravelingSalesman` del paquete `Combinatorica`, que resuelve este problema. Al usar `TravelingSalesman[G]` en un grafo G no ponderado, la instrucción

devuelve el circuito de *Hamilton* de tamaño mínimo, en función de la cantidad de aristas utilizadas. Si no existe un circuito de *Hamilton* en G , `TravelingSalesman` retorna el conjunto vacío.

Ejemplo 21



Resuelva el problema del agente viajero sobre el grafo dodecaedro.

En *Mathematica*:

```
In[ ] :=
```

```
<< Combinatorica`
```

```
G = DodecahedralGraph;
```

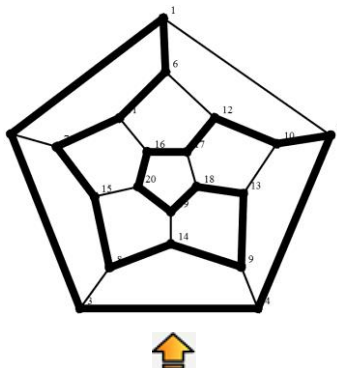
```
v = TravelingSalesman[G]
```

```
ShowGraph[Highlight[G, {Partition[v, 2, 1]}], VertexLabel -> True, PlotRange -> 0.1]
```

```
Quit[]
```

```
Out[ ] =
```

```
{1, 2, 3, 4, 5, 10, 12, 17, 16, 20, 19, 18, 13, 9, 14, 8, 15, 7, 11, 6, 1}
```



Además del problema del agente viajero, es clásico en teoría de grafos, contar con un método que determine la **longitud** y si es posible una ruta, de un **camino más corto** entre dos vértices en un grafo conexo y ponderado. A continuación, se considera este interesante tópico.



Algoritmo del camino más corto

Encontrar la **longitud de un camino más corto** entre dos nodos en un grafo conexo con pesos asociados, tiene importantes aplicaciones en distintas teorías que utilizan los grafos como un medio de representación en problemas con cierto rango de complejidad. Un algoritmo bastante conocido que resuelve este problema se denomina: **algoritmo de Dijkstra**.

Teorema 8

Sea $G=(V,E)$ un grafo conexo y ponderado, con $MP=(c_{ij})$ la matriz de adyacencia de pesos de G en el orden de los elementos de V . Si se desea obtener la longitud de un camino más corto o de peso mínimo entre dos nodos a y z de G , se realizan los siguientes pasos:

- [Inicialización] Sea $Marca(a)=0$ y $Marca(v)=\infty, \forall v, v \in V$. Sea el conjunto $LPN=V$. LPN se llama lista parcial de nodos.
- Si $z \notin LPN$, se ha terminado y $Marca(z)$ es la longitud de un camino más corto de a a z .
- Se elige un vértice w de LPN , donde $Marca(w)$ tiene el valor mínimo y se actualiza LPN como: $LPN = LPN - \{w\}$.
- La marca de cada nodo r adyacente a w tal que $r \in LPN$, se renueva así:

$$\blacksquare \text{ Marca}(r) = \text{Min}\left(\{\text{Marca}(r), \text{Marca}(w) + c_{ij}\}\right)$$

siendo c_{ij} la entrada de la matriz MP que une el vértice w con el nodo r . Si $r \notin LPN$ su marca no se actualiza. Vuelva al paso 2.

Al finalizar la ejecución del algoritmo de *Dijkstra*, la marca de z corresponde al **valor mínimo** de la suma de todos los **pesos** de las **aristas** que definen una trayectoria de a a z . El algoritmo de *Dijkstra*, por consiguiente, **no** retorna una ruta, sino un **número real**. Una implementación de este método en *Mathematica* se presenta en el ejemplo que prosigue.

Ejemplo 22



Implemente el algoritmo de *Dijkstra* con el software *Mathematica*.

La solución próxima, construye una función que recibe como parámetros el grafo G y los vértices a y z :

```

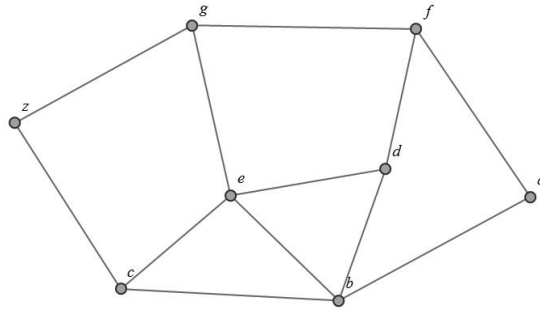
Aldijkstra[G_Graph, a_, z_] := Module[{n = VertexCount[G], TablaP =
TableForm[Normal@WeightedAdjacencyMatrix[G], TableHeadings -> {Style[#, Black] & /@
VertexList[G], Style[#, Black] & /@ VertexList[G]}]; Print["Tabla de pesos del grafo"];
Print[TablaP]; M = ConstantArray[0, {n, 2}]; T = VertexList[G]; LPN = T; L = EdgeList[G]; MP =
WeightedAdjacencyMatrix[G]; For[i = 1, i <= n, If[ToString[T[[i]]] == ToString[a], M[[i, 1]] = T[[i]],
M[[i, 1]] = T[[i]]; M[[i, 2]] = ∞ ]; i++]; Print["Inicialización: ", M]; While[MemberQ[LPN, z] == True,
mmin = M[[1, 2]]; pos = 1; For[i = 1, i <= Length[LPN], If[mmin > M[[i, 2]], mmin = M[[i, 2]]; pos = i];
i++]; vpos = Position[T, M[[pos, 1]]]; otraspos = vpos[[1, 1]]; Print["Se seleccionó el nodo: ",
M[[pos, 1]]]; For[j = 1, j <= Length[LPN], If[EdgeQ[G, M[[pos, 1]] ↔ LPN[[j]]] == True, posady =
Position[T, LPN[[j]]]; M[[j, 2]] = Min[M[[j, 2]], M[[pos, 2]] + MP[[otraspos, posady[[1, 1]]]]]; j++];
If[ToString[M[[pos, 1]]] == ToString[z], Print["La longitud del camino más corto es: ", M[[pos,
2]]]; LPN = Delete[LPN, pos]; M = Delete[M, {pos}]; Print["Lista actual de vértices: ", LPN];
Print["Lista actual de marcas: ", M]]]

```

(18)



Al correr $\text{ADijkstra}[G, a, z]$ sobre el grafo:



(19)

Cuya matriz de pesos es:

	a	b	f	c	d	e	z	g
a	0	2	1	0	0	0	0	0
b	2	0	0	2	2	4	0	0
f	1	0	0	0	3	0	0	5
c	0	2	0	0	0	3	1	0
d	0	2	3	0	0	4	0	0
e	0	4	0	3	4	0	0	7
z	0	0	0	1	0	0	0	6
g	0	0	5	0	0	7	6	0

Se obtiene:

Inicialización: $\{\{a, 0\}, \{b, \infty\}, \{f, \infty\}, \{c, \infty\}, \{d, \infty\}, \{e, \infty\}, \{z, \infty\}, \{g, \infty\}\}$

Se seleccionó el nodo: a

Lista actual de vértices: $\{b, f, c, d, e, z, g\}$

Lista actual de marcas: $\{\{b, 2\}, \{f, 1\}, \{c, \infty\}, \{d, \infty\}, \{e, \infty\}, \{z, \infty\}, \{g, \infty\}\}$

⋮

Hasta elegir el nodo z :

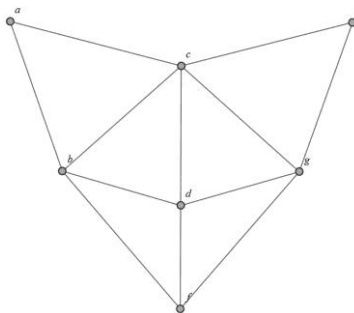
Se seleccionó el nodo: z

La longitud del camino más corto es: 5

Lista actual de vértices: $\{e, g\}$

Lista actual de marcas: $\{\{e, 6\}, \{g, 6\}\}$

Otro ejemplo, lo encontramos en el grafo:



Con ponderaciones:


	a	b	c	d	f	e	g
a	0	5	10	0	0	0	0
b	5	0	15	5	6	0	0
c	10	15	0	7	0	30	12
d	0	5	7	0	9	0	8
f	0	6	0	9	0	0	14
e	0	0	30	0	0	0	20
g	0	0	12	8	14	20	0

Si se desea hallar la longitud de un camino más corto de a al vértice g , el código (18) deriva en la respuesta:

La longitud del camino más corto es: 18

Lista actual de vértices: {e}

Lista actual de marcas: {{e, 38}}

 Dado que el algoritmo de *Dijkstra* no produce una ruta, el lector podría preguntarse cómo construir al menos un camino de longitud mínima. En este sentido, *Mathematica* tiene disponibles dos instrucciones: `FindShortestPath` y `ShortestPath`. Ambas retornan una lista de vértices que estructuran en un grafo G recibido como parámetro, una trayectoria de longitud mínima. `ShortestPath` usa el paquete `Combinatorica`.

Si se usa `FindShortestPath[G, a, z]` sobre el grafo (19), se produce la salida:

`Out[] =`

{a, b, c, z}



El estudiante puede comprobar en (19), como la trayectoria: $a \rightarrow b \rightarrow c \rightarrow z$ es de longitud mínima. Gráficamente la ruta se dibuja en *Mathematica*, así:

`In[] :=`

```
G = Graph[{a -> b, a -> f, b -> c, b -> d, b -> e, c -> e, c -> z, d -> e, d -> f, e -> g, f -> g, g -> z}, EdgeWeight -> {2, 1, 2, 2, 4, 3, 1, 4, 3, 7, 5, 6}, VertexLabels -> "Name", ImagePadding -> 10];
```

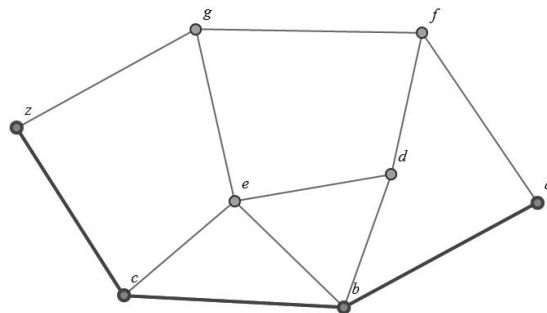
```
RutaNodos = FindShortestPath[G, a, z];
```

```
RutaAristas = Partition[RutaNodos, 2, 1];
```

```
ConstruirRuta[RutaAristas_] := Module[{L = {}}, For[i = 1, i <= Length[RutaAristas], L = Append[L, RutaAristas[[i, 1]] -> RutaAristas[[i, 2]]]; L = Append[L, RutaAristas[[i, 2]]]; i++]; L = Append[L, RutaAristas[[1, 1]]]; L = DeleteDuplicates[L]; Return[L]]
```

```
G = HighlightGraph[G, ConstruirRuta[RutaAristas], GraphHighlightStyle -> "Thick"]
```

`Out[] =`



También, *Combinatorica* incluye el comando `AllPairsShortestPath`. Esta instrucción aplicada sobre un grafo G , retorna una matriz de tamaño $n \times n$, con n la cantidad de vértices de G , donde cada entrada (i, j) es la longitud de un camino más corto entre el nodo i y el vértice j . Por ejemplo, si se utiliza el comando en el grafo bipartito completo $K_{4,6}$, obtenemos:

`In[] :=`

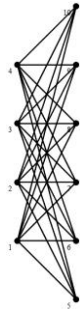
```
<< Combinatorica`
```

```
ShowGraph[G = CompleteKPartiteGraph[4, 6], VertexNumber -> True, PlotRange -> 0.1]
```

```
AllPairsShortestPath[G] // MatrixForm
```

Quit[]

Out[] =



$$\begin{pmatrix} 0 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 0 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 0 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 0 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 0 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 0 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 0 \end{pmatrix}$$

(20)

! A los lados de $K_{4,6}$ no se les asignó ningún peso, por lo tanto, las entradas de la matriz (20) son las longitudes de tamaño mínimo, encontradas de acuerdo con el número de aristas contenidas en cada trayectoria.

✶ En *Mathematica*, `GraphDistance[G, a, z]` retorna directamente la longitud de un camino más corto entre a y z . Se diferencia de la implementación (18), pues el código creado, aplica el algoritmo de *Dijkstra* **paso a paso**. En *Combinatorica* además, se facilita la setencia `Dijkstra[G, a]`, cuyo propósito consiste en formar una matriz que contiene las longitudes de uno de los caminos más cortos entre a y todos los vértices de G . Su efecto es similar a `AllPairsShortestPath`. Por ejemplo:

```

In[ ] :=
<<Combinatorica`
h = {{1, 2}, {2, 3}, {3, 5}, {4, 3}, {4, 1}, {5, 4}, {5, 2}};
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Black,
EdgeColor -> Thick], VertexLabel -> True, PlotRange -> 0.1]
G = SetEdgeWeights[G, {1, 3, 5, 8, 3, 2, 1}];
Dijkstra[G, 1][[2]]
Quit[]
Out[ ] =
{0, 1, 4, 3, 2}

```

! La indicación `[[2]]` al término de la sentencia `Dijkstra[G, 1]`, muestra como salida únicamente la segunda fila de la matriz que genera `Dijkstra[G, 1]`. El comando `Dijkstra` forma un arreglo bidimensional con dos filas, donde la primera devuelve una lista de cada uno de los padres de los vértices analizados con respecto a la dirección del nodo a . En nuestro contexto, esta fila **no interesará**.

Se comparten algunos ejercicios al lector, con la finalidad de comprobar su nivel de comprensión en esta unidad didáctica.



Ejercicios

Las soluciones de los ejercicios **más importantes** expuestos a continuación se encuentran disponibles en: <http://www.escinf.una.ac.cr/discretas>.

1. Analice qué tipo de trayectorias sobre un grafo G , devuelve el código de *Mathematica* adjunto:

```
<<Combinatorica`
```

```
h = {{1, 2}, {1, 4}, {1, 6}, {3, 4}, {2, 4}, {2, 7}, {4, 5}, {2, 3}, {2, 6}, {5, 1}, {5, 6}, {6, 4}, {6, 7}, {7, 1}};
```

```
vm = Max[h];
```

```
ShowGraph[G = SetGraphOptions[FromUnorderedPairs[h], VertexColor -> Gray, EdgeColor -> Black], VertexLabel -> True, PlotRange -> 0.1]
```

```
L = Edges[G];
```

```
CantiRutas = Length[L];
```

```
For[i = 1, i <= CantiRutas, v = L[[i]]; Ti = {v[[1]], v[[2]]}; elemento = v[[2]]; For[j = 1, j <= Length[L], w = L[[j]]; If[elemento == w[[1]], Ti = Append[Ti, w[[2]]]; elemento = w[[2]]; j++]; i++]
```

```
LR = {};
```

```
For[i = 1, i <= CantiRutas, For[j = i + 1, j <= CantiRutas, For[h = 1, h <= Length[Ti], If[Ti[[h]] == Tj[[1]], t = Join[Ti, Tj]; a = Ti[[h]]; t = Delete[t, h]; While[a != t[[2]], t = Delete[t, 2]];
```



```

If[t[[1]] == 1, LR = Prepend[LR, t]]; h++; j++; i++]
LR = DeleteDuplicates[LR];
If[MemberQ[L, {1, vm}] == True, LR = Prepend[LR, {1, vm}]]
Print[LR]

```

2. Desarrolle una función booleana en *Mathematica* que reciba un grafo G y dos de sus nodos, devolviendo True si los vértices son adyacentes y False si los nodos no se encuentran unidos por una arista en G .
3. Diseñe un programa con *Mathematica* que encuentre todas las rutas de cierta longitud, leída por teclado, sobre un grafo G .
4. Responda a las siguientes preguntas con ayuda de *Mathematica*: ¿es válido el teorema 2 si G es un grafo con lazos?, ¿qué ocurre si G posee lados múltiples?
5. Elabore una rutina en *Mathematica*, donde siendo m la cantidad de aristas de un grafo completo K_n , se verifique:

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

6. Conjeture si es posible, una fórmula para la longitud máxima de todas las rutas en un grafo completo de n nodos ¿Es esto posible en un grafo bipartito completo?
7. Sea G un digrafo. Determine con ayuda de *Mathematica*, la validez del teorema 3.
8. Modifique la función (12), sustituyendo el comando `AdjacencyMatrix` por `ToAdjacencyMatrix`.
9. Represente por medio de la matriz de adyacencia e incidencia en distintos órdenes escogidos por el usuario, los grafos: rueda, estrella, círculo, camino, cuadrícula y regular, ¿se observa alguna propiedad? Explique.
10. Encuentre el número mínimo de aristas y de nodos que deben tener los grafos: completo, bipartito completo, rueda, estrella, círculo, camino, cuadrícula y regular, con la finalidad de continuar siendo conexos. Sugerencia: utilice los comandos `EdgeConnectivity` y `VertexConnectivity`.
11. Represente los siguientes grafos en *Mathematica*, dada su matriz de adyacencia

de pesos (sugerencia: completar con ∞ cuando dos nodos no están relacionados y utilizar la instrucción `WeightedAdjacencyGraph`):

a.
$$\begin{pmatrix} 0 & 1 & 0 & 3 & 5 \\ 1 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 3 & 1 \\ 3 & 4 & 3 & 0 & 2 \\ 5 & 1 & 1 & 2 & 0 \end{pmatrix}$$

b.
$$\begin{pmatrix} 0 & 5 & 2 & 3 & 1 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 & 1 & 0 \\ 2 & 3 & 0 & 0 & 2 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 8 \\ 1 & 2 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 6 & 0 & 0 & 0 \\ 0 & 0 & 1 & 8 & 1 & 3 & 0 \end{pmatrix}$$

12. Conjeture con ayuda del *software Mathematica*, un teorema similar al expuesto en la página 59, suponiendo que el grafo G es dirigido.
13. Elabore una rutina en *Mathematica* que determine si un grafo que puede contener un ciclo posee un circuito de *Euler*.
14. Diseñe una implementación en *Mathematica* del algoritmo de *Fleury* distinta a la presentada en la página 63, para encontrar una ruta de *Euler* en un grafo que la contenga.
15. Con ayuda de *Mathematica* responda: ¿cuál condición debe cumplir K_n para tener un circuito de *Hamilton*?
16. Los grafos: rueda, estrella, círculo, camino, cuadrícula y regular, ¿pueden contener circuitos de *Euler* o de *Hamilton*? ¿Qué condiciones deben satisfacer en caso afirmativo?
17. Programe en *Mathematica* una solución paso a paso del problema del agente viajero.
18. Modifique el programa (18) con la intención de devolver una ruta de longitud mínima entre dos vértices, en un grafo conexo y ponderado.
19. Construya una función que genere de forma pseudoaleatoria un grafo no simple con n nodos y con un máximo de m lados múltiples



Bibliografía

- Ávila, J. (2005). *Estructuras de matemática discreta para computación*. Costa Rica: UNA.
- Caballero, M., Migallón, V. & Penadés, J. (2001). *Prácticas de matemática discreta con MAGRADA*. España: Publicaciones Universidad de Alicante.
- Castro, I. (1988). *El más prolífico en la historia de la matemática: Leonhard Euler*. Colombia: Pontificia Universidad Javeriana.
- Grassman, W. & Tremblay, J. (1997). *Matemática discreta y lógica*. España: Prentice-Hall.
- Hastings, K. (2006). *Introduction to the mathematics of operations research with Mathematica*. USA: Chapman & Hall/CRC.
- Johnsonbaugh, R. (2005). *Matemáticas discretas*. México: Pearson Prentice Hall.
- Kolman, B., Busby, R. & Ross, S. (1995). *Estructuras de matemáticas discretas para computación*. México: Prentice-Hall Hispanoamericana.
- Maeder, R. (2000). *Computer science with Mathematica: theory and practice for science, mathematics, and engineering*. USA: Cambridge University Press.
- Lipschutz, S. & Lipson, M. (2004). *2000 problemas resueltos de matemática discreta*. España: Mc. Graw-Hill.
- Pemmaraju, S. & Skiena, S. (2009). *Computational discrete mathematics: combinatorics and graph theory with Mathematica*. USA: Cambridge University Press.

- Rosen, K. (2007). *Discrete mathematics and its applications*. USA: Mc. Graw-Hill.
- Shiflet, A & Shiflet, G. (2006). *Introduction to Computational Science: Modeling and Simulation for the Sciences*. USA: Princeton University Press.
- Vílchez, E. (2015). *Estructuras Discretas con Mathematica*. México: Alfaomega.

CONCLUSIONES

El proyecto ya expuesto ha permitido derivar una serie de conclusiones obtenidas después de un proceso de reflexión, análisis y síntesis, que se comparten a continuación:

1. En la Universidad Nacional de Costa Rica existen pocas propuestas pedagógicas a nivel de grado, relacionadas con cursos con un formato bimodal. El presente proyecto ha aportado a sufragar estas falencias, a través de una formulación pionera en el campo de la matemática discreta.
2. Los docentes de la cátedra del curso *EIF-203 Estructuras Discretas para Informática* poseen a nivel general, poca experiencia con respecto a la implementación de estrategias de enseñanza apoyadas en entornos virtuales de aprendizaje. La presente propuesta ha contribuido con un enfoque innovador, homogéneo y sistemático que permitirá en futuras versiones de la materia incorporar a todos los participantes de la cátedra, con la intención de promover e impulsar experiencias educativas basadas en un enfoque bimodal.
3. La Escuela de Informática de la Universidad Nacional de Costa Rica, reconoce en una visión de futuro la necesidad de transformar sus prácticas de enseñanza y aprendizaje, dando cabida al uso de ecosistemas digitales como mecanismos de flujo para preparar a futuros ingenieros en sistemas de información, no solamente en conocimientos de naturaleza científica, sino también, en el aprendizaje de habilidades y destrezas relacionadas con competencias de la era digital, tales como: la validación de fuentes, la generación de inferencias, la transformación, uso y aplicabilidad del conocimiento, la autonomía y el aprender a aprender.
4. El entorno de aprendizaje virtual desarrollado como producto principal de este proyecto reconoce:
 - a. La importancia de un proceso de planificación serio hacia la consolidación de nuevos formatos educativos, que sirvan de complemento a la presencialidad.
 - b. La integración del uso de las tecnologías digitales como recursos que pueden favorecer positivamente el aprendizaje de conocimientos matemáticos, cuando han sido plasmados en el marco de acción de una propuesta pedagógica clara y responsable.

- c. La necesaria transformación didáctica en las instituciones de enseñanza superior a través de estrategias metodológicas centradas en el alumno y en las conexiones que ellos conforman dentro los procesos de aprendizaje.
 - d. La importancia de la adquisición de conocimientos tácitos en la población estudiantil como recursos para el tratamiento de la incertidumbre característica de los mercados laborales actuales, que demandan personas capaces de aprender a lo largo de la vida y en compañía de otros.
5. El curso *EIF-203 Estructuras Discretas para Informática* bajo un enfoque bimodal constituye una iniciativa pedagógica en un campo de conocimiento árido y de naturaleza fáctica, pretendiendo incentivar una permutación en los roles asumidos típicamente por profesores y alumnos en cursos de matemática a nivel universitario. El proyecto hace la invitación expresa de favorecer escenarios educativos donde prevalezca la observación, el análisis, la resolución de problemas, la creatividad, el descubrimiento y la puesta en ejecución de formas de razonamiento personales nutridas por la riqueza que provee el trabajo colaborativo y su consecuente búsqueda de conceso.
6. La propuesta de diseño y desarrollo del entorno de aprendizaje virtual para la materia *EIF-203 Estructuras Discretas para Informática* representa un modelo que otras cátedras de los cursos de la carrera de *Ingeniería en Sistemas de Información* de la UNA, pueden adoptar a futuro, hacia la conformación de un nuevo plan de estudios que favorezca formalmente otros enfoques bimodales.